Maximum-stability Distributed Control in Traffic Networks

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Rongsheng Chen

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE THE DEGREE OF
DOCTOR OF PHILOSOPHY

Adviser: Michael W. Levin

May 2021

© Rongsheng Chen 2021

ACKNOWLEDGEMENTS

I want to express my gratitude to my advisor, Professor Michael Levin, for his consistent guidance, support, and encouragement throughout my Ph.D. study. I got so much inspiration and power to reach my goal in the area of transportation. Without his help, I could not finish my research for my Ph.D. degree and could not make so much progress during my Ph.D. study. I would also like to thank Professor Gary Davis, Professor Alireza Khani, and Professor Kevin Leder for serving on my exam committee and giving me constructive comments on my research.

I want to thank all of my friends in Minnesota for the company in these six years. The winter in Minnesota is cold, but the friendship is warm. No matter where I will be in the future, I will never forget the happy memory I have in Minnesota.

At last, I would like to thank my parents for their love and support. They are my strong backing, and they teach me to be a person that always looks upon the bright side of things.

ABSTRACT

The max-pressure control is a distributed control algorithm that has the property of stabilizing the total queue length in the network theoretically. In spite of its good properties, some assumptions or requirements of the max-pressure control make it hard to be applied to traffic networks in reality: such as the data collection of queue length information for each movement and fixed route choices. Besides, traditional max-pressure control algorithms are only designed for signal-controlled intersections and are not applicable for signal-free intersections. Therefore, this thesis proposes max-pressure control algorithms and tests their performances in traffic networks while relaxing some of the assumptions used in existing studies. This thesis first explores mild assumptions for weight functions to incorporate alternative data sources in max-pressure control. This thesis also proposes an autonomous intersection management (AIM) algorithm considering pedestrians using the max-pressure control. Besides, the performance of max-pressure control is tested when road users' route choice is considered using dynamic traffic assignment, and a routing guidance algorithm is also developed to modify road users' route choices and to improve network efficiency.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Traffic congestion is one of the significant problems in the urban traffic system. The slow travel speed increases the travel time and reduces the progress of road users. In traffic congestion, vehicles need to accelerate and decelerate frequently, which increases the greenhouse gas emissions and air pollution in the city. In 2017, the traffic congestion in peak hours resulted in a total cost of $305 billion to U.S. drivers, which is equivalent to a cost of $1,445 to each driver (Pyzyk, 2018). Traffic control plays a vital role in enhancing traffic safety and mitigate traffic congestion in the network. There are many applications of traffic control, such as ramp metering and speed limits on freeways, and traffic signals at intersections. For urban traffic, intersections are the main bottlenecks and contribute to most delays of vehicles. Reducing vehicle delays at intersections will significantly improve traffic efficiency in traffic networks. in this thesis, intersection control is studied to improve traffic efficiency and to mitigate traffic congestion in urban road networks.

1.1 Intersection control

The basic principle of intersection control is to separate vehicles from different directions temporally or spatially to avoid conflicts within an intersection. Intersection control starts from the stop-sign control based on the rule of first-come-first-served. Traffic lights later

serve as the indicator of the right of way. Vehicles in specific directions are allowed to move when the light turns green for them. In fixed signal timings, the lengths of green light intervals are calculated in advance, which needs to be manually calibrated to traffic data. The signal timings of the fixed traffic signal for different times in a day are usually different, but they are only updated every 3–5 years. With the development of traffic detectors, real-time traffic data collected by detectors, such as loop detectors, video cameras, and radar detectors, provides the information for intersection control. Based on real-time traffic data, adaptive intersection control updates signal timings at isolated intersections or interconnected intersections on an arterial or in an area. Some frameworks of adaptive signal control have also been developed and applied to city-wide networks, such as SCOOT (Hunt et al., 1981), SCATS (Sims and Dobinson, 1980), RHODES (Mirchandani and Head, 2001), OPAC (Gartner, 1983), and etc.

Traditional traffic detectors have a limited detection range because they are installed at fixed points, so they are categorized as Eulerian sensing (Xia et al., 2017), whose observations come from a fixed observation location. The disadvantages of these detectors are obvious: the inductive-loop sensors are easy to break, and the video cameras have low accuracy during adverse weather. The new communication technologies, such as vehicle-to-vehicle (V2V), and vehicle-to-infrastructure (V2I) technologies, bring new methods to collect traffic data. These detectors capture the trajectories of observations moving with traffic flow, which is known as Lagrangian sensing. Advances in vehicular communication technologies bring the evolution of intersection controls because of more available traffic data, and intersections that are not equipped with traffic detectors can also implement adaptive intersection control. For traditional adaptive intersection control, existing algorithms that use vehicle trajectory data can optimize the order and the activation time of phases. With the rapid development of autonomous driving technologies in recent years, there are also algorithms proposed for signal-free intersection control to coordinate non-conflicting trajectories at the intersection (Dresner and Stone, 2004) and to adjust driving speeds or

accelerations of vehicles based on existing traffic signals so that vehicles can pass the intersection smoothly and avoid stopping for the red light (Ma et al., 2017).

1.2 Centralized and distributed controls

In most cases, controlling an isolated intersection may not have a significant effect in mitigating congestion for a network. When the queue at a downstream intersection is too long and extends to upstream intersections, queue spillback occurs, which may lead to gridlock of the network. Only after the queue at the downstream intersection is cleared can the queue at the upstream intersection be cleared. Coordinated traffic control allows several intersections on an arterial or in a zone to cooperate so that vehicles can smoothly pass through multiple intersections without stopping for red lights. When multiple intersections are coordinated, simple intersection control systems are preferred because applying complicated intersection controls to a large area is expensive and difficult to maintain.

With the expansion of the city size and the increase of traffic volumes on the road, more and more complicated algorithms are developed to compute ”optimal” signal controls for intersections using various data sources. When the algorithm is applied to a city-wide network, it needs to coordinate a large number of intersections. The coordination between intersections should be supported by an appropriate control structure. There are two main types of control structures: centralized control and distributed control.

1.2.1 Centralized control

In a centralized control system, a central controller makes all decisions, so it needs the information of the entire system. If there is a traffic network using a centralized control system, the central controller that connects with all intersections in the area needs to collect the traffic data from all intersections. After the data collection, the central controller constructs models and determines all controls for intersections, which is able to perform

a global task, such as minimizing the total vehicle delays. Most frameworks of adaptive control systems use centralized control. Centralized control systems is not always used in a large network, but can also be applied to small regions. Individual corridors can use centralized control system to coordinate intersections along the corridor and help vehicles pass through the corridor smoothly.

This control structure needs stable connectivity to ensure information communication, which restricts the size of the system. In addition, a malfunction of the controller can lead to the breakdown of the entire system as the central controller makes decisions for all nodes.

1.2.2 Distributed control

Unlike a centralized control system, a distributed control system has multiple controllers. Each controller has its own control area and collects data from a limited number of nodes. Each controller makes its own decision but the interaction between controllers is admitted. Aggregating all decisions of controllers makes up the decision of the entire system. For a distributed control system, it is not as easy to reach global optimality as a centralized control system because each controller operates independently, unless the system structure is well-designed.

The ideal condition for a traffic network using a distributed control system is that controllers at each intersection collect the traffic data from upstream or downstream intersections and compute the intersection control by themselves, and minimize the total delay or maximize the total throughput of the network.

With the expansion of cities and the increase of traffic volumes, intersection control algorithms should be applied to more and more intersections. Comparing centralized control systems and distributed control systems that can perform global tasks, centralized control systems may face scalability problems in terms of computation time and model complexity.

Distributed control needs less information transmission between intersections and smaller computation time, which is more suitable to large networks.

1.3 Max-pressure control

This thesis focuses on applying a distributed controller called the max-pressure controller to traffic networks. As a distributed control algorithm, it is suitable for controlling a large network, in which each intersection uses the traffic data from the current intersection and adjacent intersections and calculates the optimal intersection control by itself. It is also proven to maximize the total throughput and stabilize the total queue length of the network theoretically (Varaiya, 2013; Le et al., 2015).

The max-pressure control was originally used as a scheduling policy in communication and power networks (Tassiulas, 1992), and was originally called "back-pressure" control. This control policy was used to activate servers to serve specific queues at nodes in multi-hop radio networks. It was first applied in traffic networks in two studies (Wongpiromsarn et al., 2012; Varaiya, 2013). These two studies used "back-pressure" (Wongpiromsarn et al., 2012) and "max-pressure" (Varaiya, 2013) respectively, and their algorithms mainly differ in the queuing model and the way they defined the weight or pressure of a turning movement. In this thesis, the word "max-pressure" is used to represent both types of controls.

A movement is the basic unit in the model for the max-pressure control, it is defined as a pair of lanes or links, for example, (i, j). The weight of each movement in most existing studies is defined as follows:

$$w_{ij}(t) = x_{ij}(t) - \sum_{k \in \Gamma_j^+} x_{jk}(t) p_{jk}(t) \tag{1.1}$$

The weight of movement (i, j) is the queue length of movement (i, j) minus the average

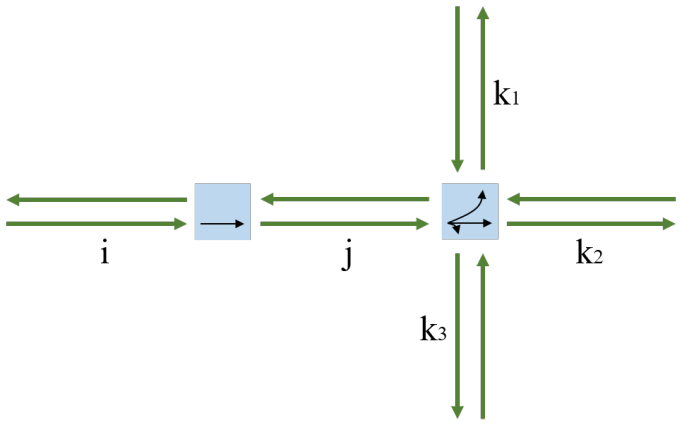


Figure 1.1: Downstream movements

queue length of movements on downstream links $k \in \Gamma_j^+$. For example, in Figure 1.1, the downstream movements of movement (i, j) are movements (j, k_1), (j, k_2), and (j, k_2). A traffic signal phase is the time when multiple vehicle turning movements can move. The objective is maximizing the total value of the term $Q_{ij}w_{ij}(t)s_{ij}(t)$ for all turning movements, where Q_{ij} is the capacity of the turning movement (i, j) and $s_{ij}(t)$ is the signal activation of the turning movement (i, j) at time t. The controller at each intersection calculates the optimal intersection control that maximizes the objective function of the intersection in (1.2), where s_n is the intersection control for intersection n and $\mathcal{S}_n$ is a set of intersection control.

$$s^\star(t) = \arg\max_{s_n \in \mathcal{S}_n} \left\{ \sum_{(i,j)} Q_{ij}w_{ij}(t)s_{ij}(t) \right\} \tag{1.2}$$

The general process of using the max-pressure control for each intersection is:

1. Collect traffic information (for example, queue length information)

2. Calculate the weight or pressure of each movement and the weight of each phase

3. Activate the phase with the maximum pressure value for a fixed time interval or

assign green times proportionally to each phase in a cycle

1.4 Problem statements and contributions

Although the max-pressure control has some good properties theoretically, some assumptions or requirements of the max-pressure control make it hard to be applied in traffic networks in reality: such as the data collection of queue length information for each movement and fixed route choices. Besides, traditional max-pressure control algorithms are only designed for signal-controlled intersections and are not applicable for signal-free intersections. Therefore, the objective of this thesis is to propose max-pressure control algorithms and test their performances in traffic networks while relaxing some of the assumptions used in existing studies. This thesis mainly includes the following four topics.

1.4.1 Using travel time in the max-pressure control

The first topic looks for a replacement for the queue length information used by the max-pressure control. The data collection of queue length information requires loop or video detectors to be installed at each approach. Data quality greatly affects the efficiency of max-pressure control. However, sensors require maintenance, and their accuracy is limited by adverse road or weather conditions. If max-pressure controls can use reliable online traffic information which does not require the installation of sensors in the field, the cost of implementing the max-pressure control will be greatly reduced and there will be a larger possibility for the max-pressure control to be used at intersections without sensor infrastructure investments. As traditional max-pressure control uses queue lengths and its queue length stability has been proven in many studies (Varaiya, 2013; Xiao et al., 2014), an inevitable problem in using travel times is that it is uncertain if max-pressure control can maintain queue length stability. Ideally, the travel time function should represent the available data as accurately as possible. Simultaneously, the travel time function should

be carefully designed so that it maintains the analytical stability properties of the max-pressure control so it is necessary to find sufficient yet realistic assumptions for the travel time function to retain favorable stability properties.

1.4.2 Integration of autonomous intersection management with max-pressure control considering pedestrian access

The second topic tries to integrate the max-pressure control with the autonomous intersection management (AIM) algorithms (Dresner and Stone, 2004), which represent signal-free intersection control algorithms to coordinate non-conflicting trajectories of vehicles at the intersection. With the advances in autonomous vehicle technology, vehicles can be precisely controlled by computers. Once vehicle trajectories are determined, vehicles can follow these assigned trajectories and avoid collisions without the safety buffers of traffic signal phases.

Most AIM models do not consider pedestrian access, but pedestrians may still require intersection access due to the costs of constructing separate right-of-way for pedestrians (e.g. tunnels or bridges) in the future. Having pedestrians at intersections controlled by AIM brings a lot of unpredictable risks. If an AIM-controlled intersection calculates vehicle trajectories only based on vehicle information and does not consider pedestrians, the calculated vehicle trajectories are not compatible with intersection access for pedestrians. To minimize the vehicle delay at the intersection, AIM models leave small gaps between vehicles. It is hard for pedestrians to find a safe gap for them to cross the street. The detectors on autonomous vehicles enable vehicles to react to jaywalking pedestrians, but the resulting unplanned stop causes the temporary breakdown of the intersection traffic. Therefore, it is reasonable to add crosswalk activation in AIM to incorporate pedestrians in intersection control. When one or multiple crosswalks are activated, vehicles whose trajectories do not intersect with the activated crosswalks are allowed to move. As acti-

vating crosswalks in AIM reduces the throughput of vehicles at the intersection because it blocks vehicles with conflicting trajectories, the max-pressure control can be applied here to maximize the network throughput of the total combined vehicle and pedestrian flow.

1.4.3 User equilibrium analysis with max-pressure control

The third topic explores the performance of max-pressure control when the route choice of road users is not fixed. Most existing studies assume a constant average turning proportion from one link to another when designing and testing the effect of max-pressure control, which indicates that route choice behaviors of road users are fixed regardless of travel times. However, this assumption may not be realistic when road users react to the intersection control by picking the path with the smallest travel time, which is composed of the travel time on the link and waiting time at the intersection. When intersection control is used in practice, the route choice behavior of road users may offset its control effectiveness. For example, the efficiency of a fixed-time traffic signal can be greatly affected if most travelers switch from an entry lane with a long green light time to another entry lane with a short green light time. With travelers' route choice behaviors, the performance of an intersection control algorithm may not be as good as it is expected.

Traffic assignment is used to analyze the performance of max-pressure control under varying route choices. It is the last step in the four-step method, which determines the path that each traveler uses to reach the destination and calculates the traffic flow or the travel time on each road. The result of the traffic assignment helps traffic planners to find the bottleneck of a traffic network or evaluate the performance of a traffic control algorithm. Moreover, it is unknown if there is a user equilibrium in a network with max-pressure control. The user equilibrium is defined as the state when no traveler in the network can reduce his or her travel time by unilaterally changing his or her path. The existence of the user equilibrium has been validated with real traffic networks for many other intersection

control algorithms but not for the max-pressure control. Therefore, using dynamic traffic assignment and calculate the route distribution of travelers, we can test the performance of the max-pressure in a realistic way. Besides, it is also worth testing the effects of model convergence on the performance evaluation when there are differences in the travel cost of used paths.

1.4.4 Routing guidance algorithm based on the max-pressure control

The fourth topic is interested in the application of the distributed controller on modifying road users' route choices. Road users' route choice and intersection control are two main factors that affect the efficiency of the traffic network. These two factors also have an interactive relation. For example, the signal timing of fixed-time controllers is calculated based on the traffic volumes of traffic movements passing through the intersection, such as the Webster's method (Webster, 1958). After implementing the calculated signal timing, the route choice of road users may change based on new experienced travel times. The modified route choice may result in worse overall network performance (Smith, 2015). If intersections are able to affect the travelers' route choice, it can possibly improve the overall performance of the network. For human-driven vehicles, it is nearly impossible to control route choices. For autonomous vehicles, this task is much easier. Besides, it is interesting to explore if the network performance under the control of the proposed algorithm can be better than the user equilibrium motivated by travelers' self-interested choices, which stand for a realistic result of traffic assignment when each traveler chooses the shortest path.

The relation between the four research topics is shown in Figure 1.2. The first two topics are based on the assumption of fixed route choice behaviors, while the last two topics are based on varying route choice behaviors. The first topic explores a replacement of queue length information as the input of the max-pressure control. The third topic evaluates the performance of the max-pressure control following Wardrop's first principle. Both the first

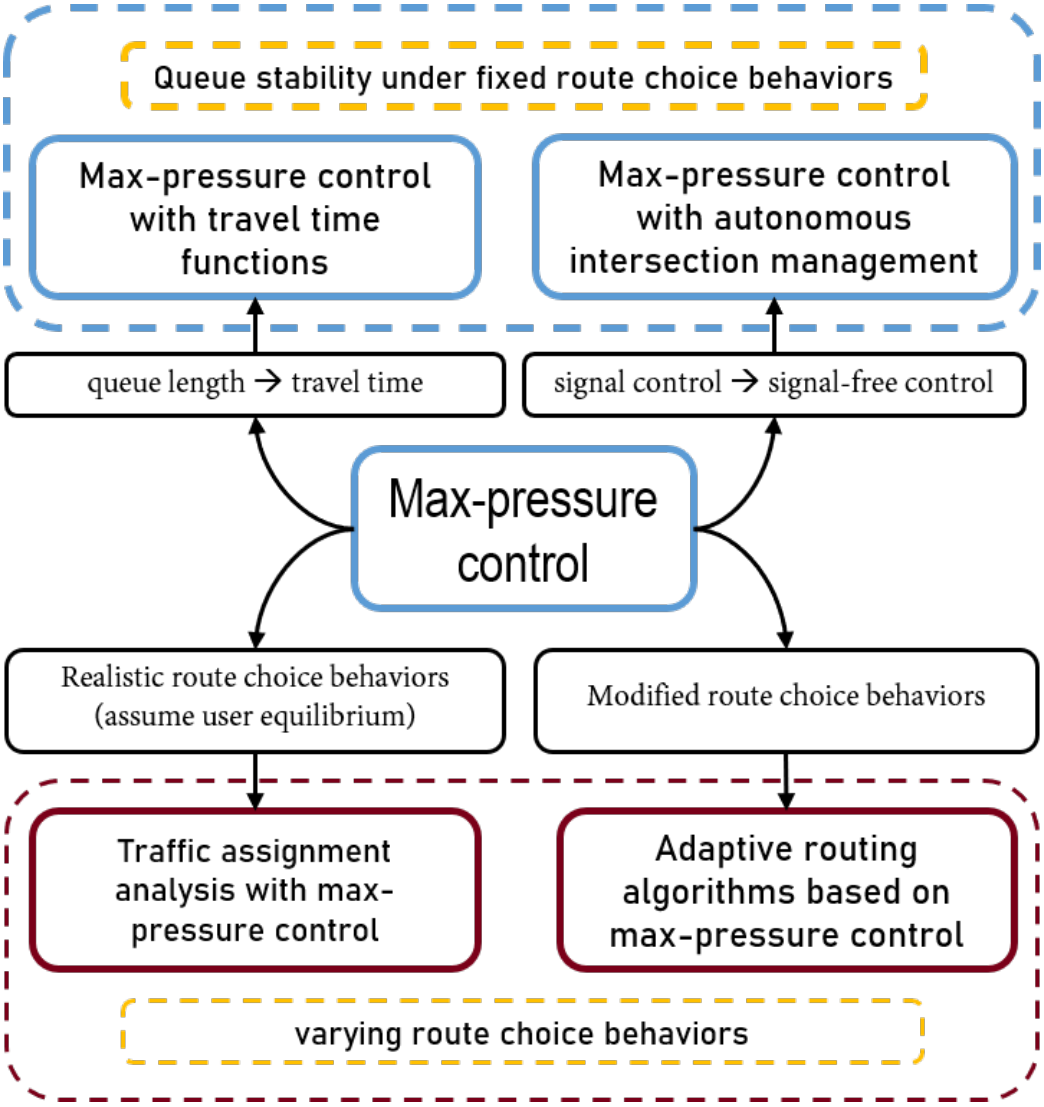


Figure 1.2: Research topics

topic and the third topic assume traffic networks with human-driven vehicles. The second topic converts signal control to signal-free control for autonomous vehicles. The fourth topic explores algorithms to modify the route choice of road users. Both the second topic and the fourth topic assume traffic networks with autonomous vehicles.

This thesis mainly focuses on answering the following questions related to mentioned four topics.

1. What form of travel time functions can be used for the max-pressure control and maintain network-level properties of the max-pressure control?

2. How can the max-pressure control be integrated with intersection controls for autonomous vehicles and consider pedestrians on the road?

3. How do route choice behaviors affect the performance of max-pressure control in a large network?

4. How can the max-pressure control be applied to route vehicles and improve the efficiency of traffic networks?

1.5 Thesis organization

The remainder of the thesis includes these chapters: Chapter 2 briefly introduces the relevant literature about the max-pressure control, the autonomous intersection management, traffic assignment problem with intersection control, and the routing guidance algorithms with max-pressure control. Chapter 3 includes the network model, the signal control, and the definition of the stability region. Chapter 4 illustrates the assumptions that travel time functions should follow to maintain the theoretical network queue length stability. It also shows the analytical results and the results of numerical experiments. Chapter 5 proposes an AIM algorithm with max-pressure control considering the pedestrian access and proves

the stability of the proposed algorithm. Chapter 6 uses the dynamic traffic assignment to evaluate the performance of max-pressure control. Chapter 7 prepares a network in a microscopic simulation software and compared the performances of different types of max-pressure control with other distributed controllers in a network with realistic settings. Chapter 8 proposes a routing guidance algorithm based on the max-pressure control.

Chapter 2

Literature Review

This chapter introduces the existing studies related to relevant topics, including the formulation for the weight and the pressure, the validation for the network-level properties of the max-pressure control, autonomous intersection control, traffic assignment model considering the intersection control, and routing algorithm with max-pressure control.

2.1 Weight/pressure functions in max-pressure control

Among studies that applying the max-pressure control to traffic networks, two types of queuing models are used, which calculate the queue evolution for a link (Wongpiromsarn et al., 2012) and for a turning movement (Varaiya, 2013) respectively, as shown in equations (2.1) and (2.2a). In equation (2.1), $x_i(t)$ is the queue length on link i at time t, $I_i(t)$ is the number of arriving vehicles at time t, and $O_i(t)$ is the number of exiting vehicles at time t. In equation (2.2a), $x_{ij}(t)$ is the queue length of movement (i, j), $y_{ij}(t)$ is the number of vehicles exiting the queue of movement (i, j), and $d_i(t)$ is the number of arriving vehicles on entry link i, which is assumed to be independent identically-distributed random variables with respect to t. $p_{ij}(t)$ is the proportion of vehicles on link i going to j, whose average value is assumed to be fixed.

$$x_i(t+1) = x_i(t) + I_i(t) - O_i(t) \tag{2.1}$$

$$x_{ij}(t+1) = x_{ij}(t) - y_{ij}(t) + \sum_{h \in \Gamma_i^-} y_{hi}(t) p_{ij}(t) \quad \forall i \in \mathcal{L}_{\text{int}}, h \in \Gamma_i^-, j \in \Gamma_i^+ \tag{2.2a}$$

$$x_{ij}(t+1) = x_{ij}(t) - y_{ij}(t) + d_i(t) p_{ij}(t) \quad \forall i \in \mathcal{L}_{\text{entry}}, j \in \Gamma_i^+ \tag{2.2b}$$

As queue models determine the form of weight functions, there are two forms of weight functions used in the literature. Some studies follow the form of the weight function in Wongpiromsarn et al.'s study, which is the difference of pressure functions on the upstream and downstream links of a turning movement (Gregoire et al., 2014a,b; Zaidi et al., 2016). As shown in equation (2.3), $x_i(t)$ and $x_j(t)$ are queue lengths or other traffic states for link i and its downstream link j respectively. This form is used in most studies that call their control policies as "back-pressure control". In these studies, the pressure of the turning movement is first determined, then the weight of each phase is determined. Gregoire et al. (2014a) used a linear pressure function with a strictly positive slope. Gregoire et al. (2014b) later designed a convex normalized pressure function.

$$w_{ij}(t) = x_i(t) - x_j(t) \tag{2.3}$$

The second type of weight function was first proposed by Varaiya (2013). He formulated the weight as the difference between the queue of the current turning movement and the weighted average queue length of downstream turning movements, which is shown in equation (1.1). In equation (1.1), $x_{ij}(t)$ and $x_{jk}(t)$ are queue lengths or other traffic states of turning movement (i, j) and downstream turning movement (j, k) respectively. Many studies follow the same form of the weight function in Varaiya's study (Xiao et al., 2014; Le et al., 2015; Pumir et al., 2015; Le et al., 2017; Hsieh et al., 2017; Rey and Levin, 2019; Lioris et al., 2016; Kouvelas et al., 2014), which is the traffic state of the current turning movements minus the weighted average traffic state of downstream turning movements. The pressure calculation is after the weight calculation of each link or turning movement. Xiao et al. (2014) proposed a weight function that included the capacity of the

turning movement. Le et al. (2015) and Pumir et al. (2015) both developed cyclic phase max-control policies. Le et al. (2015) proportionally assigned the available green time to predefined phases based on their weights and Pumir et al. (2015) solved a linear program with a constraint of minimum activation time to obtain the optimal control. Hsieh et al. (2017) considered the switch-over delay between phases. Le et al. (2017) incorporated a utility function that was able to react to drivers' route choices. Then the optimal signal control was determined by solving a minimization problem including both the utility function and the weights of available phases. Li and Jabari (2019) designed a position weighted max-pressure control policy. The weight function needed the input of the traffic density distribution of a link. Rey and Levin (2019) proposed an intersection control algorithm that integrated autonomous intersection management (AIM) with the max-pressure control considering both autonomous and legacy vehicles. Table 2.1 shows different weight and pressure functions in existing studies. To help with the understanding of the form of these weight and pressure functions, notations in difference papers are unified and the unified notations are shown in Table 2.2.

Table 2.1: Functions used to calculate the weight and the pressure (unified notations)

study	weight	pressure	proved stable
Wongpiromsarn et al. (2012)	$w_{ij} = x_i(t) - x_j(t)$	$w_{ij}(t)\xi_i(p, \mathcal{L}_i, \mathcal{L}_i, z_i(t))$	Y
Varaiya (2013)	$w_{ij} = x_{ij} - \sum_{k\in\Gamma_j^+} x_{jk}p_{jk}$	$Q_{ij}w_{ij}s_{ij}$	Y
Xiao et al. (2014)	/	$P_{ij}(t) = \frac{\alpha_{ij}\left[\sum_{(i',j')\in\mathcal{L}_{ij}^{(\text{in})}} x_{i'j'}(t)\right]}{Q_{ij}} x_{ij}(t)$	Y
Gregoire et al. (2014a)	$w_{ij}(t) = \min(x_{ij}(t)/Q_{ij}, 1)\max(P_i(t) - P_j(t), 0)$	$P_i(t) = \sum_{j\in\Gamma_i^+} P_{ij}(x_{ij}) = \sum_{j\in\Gamma_i^+} \theta_{ij}x_{ij}$	N
Le et al. (2015)	$w_{ij} = x_{ij} - \sum_{k\in\Gamma_j^+} x_{jk}p_{jk}$	/	Y
Gregoire et al. (2014b)	$w_{ij} = \max(P_i(t) - P_j(t), 0)$	$P_i(x_i) = \min\left(1, \frac{\frac{x_i}{Q_\infty} + \left(2 - \frac{x_i}{Q_\infty}\right)\left(\frac{x_i}{Q_i}\right)^m}{1 + \left(\frac{x_i}{Q_i}\right)^{m-1}}\right)$	Y
Pumir et al. (2015)	$w_{ij} = x_{ij} - \sum_{k\in\Gamma_j^+} x_{jk}p_{jk}$	$Q_{ij}w_{ij}s_{ij}$	Y
Zaidi et al. (2016)	$w_{ij}(t) = \max\{x_i(t) - x_j(t), 0\}$	$w_{ij}(t)s_{ij}(t)$	N
Le et al. (2017)	$w_\psi(t) = \sum_{i\in\Gamma_j^-} \psi_i(x_i(t) - \sum_{j\in\Gamma_i^+} \bar{\theta}_{ij}(t)x_j(t))$	/	Y
Hsieh et al. (2017)	$w_{ij} = x_{ij} - \sum_{k\in\Gamma_j^+} x_{jk}p_{jk}$	$Q_{ij}w_{ij}s_{ij}$	Y
Wu et al. (2017)	$w_{ij}(t) \triangleq T_{ij}(t)$	$\gamma_{ij}w_{ij}(t)s_{ij}$	Y
Li and Jabari (2019)	$w_{ij}(t) = f(l_i, l_j, c_{ij}, \pi_{ij}, \delta_{ij}, \theta_j)$	$w_{ij}(t)\mathbb{E}^{\rho(t)}q_{ij}(p)$	Y
Rey and Levin (2019)	$w_{ij} = x_{ij} - \sum_{k\in\Gamma_j^+} x_{jk}p_{jk}$	$w_i(t)y_i(t)$	Y
Lioris et al. (2016)	$w_{ij} = x_{ij} - \sum_{k\in\Gamma_j^+} x_{jk}p_{jk}$	$Q_{ij}w_{ij}s_{ij}$	N
Kouvelas et al. (2014)	/	$P_i(t) = \left[\frac{x_i(t)}{x_{i,\max}} - \sum_{j\in O_n} \frac{\beta_{ij}x_j(t)}{x_{j,\max}} S_i(t)\right] Q_i, i \in I_n$	N

Table 2.2: Notations in Table 2.1

Study	Notation	Represent
	w_{ij}	weight of turning movement (i, j)
	Q_i, Q_{ij}	capacity of link i or turning movement (i, j)
	p_{ij}	the proportion of vehicle on link i going to link j
	x_i, x_{ij}	queue length on link i or turning movement (i, j)
	s_{ij}	the activation of turning movement (i, j) or link i
	P_i, P_{ij}	pressure of link i or turning movement (i, j)
	$\xi_i(p, \mathcal{L}_i, \mathcal{L}_i, z_i(t))$	service rate for movement (i, j) (Wongpiromsarn et al., 2012)
(Le et al., 2017)	ψ, ψ_i	a phase, the activation of link i in phase ψ
(Gregoire et al., 2014b)	Q_∞	the slope of the pressure at low occupancy
(Wu et al., 2017)	γ_{ij}	a positive constant to give more emphasis to certain movements
(Li and Jabari, 2019)	$f(l_i, l_j, c_{ij}, \pi_{ij}, \delta_{ij}, \theta_j)$	a function using the traffic state distribution along the road
(Li and Jabari, 2019)	$\mathbb{E}^{\rho(t)}q_{ij}(p)$	the expected fluxes for movement (i, j)
(Wu et al., 2017)	$T_{ij}(t)$	a delay function of turning movement (i, j)
(Rey and Levin, 2019)	y_{ij}	the traffic flow of movement (i, j)
(Kouvelas et al., 2014)	I_n, O_n	the incoming and outgoing link set of intersection n
(Xiao et al., 2014)	$\mathcal{L}_{ij}^{(\text{in})}$	a set of turning movement (i', j') on the same path as (i, j) and link i' should be an entry link

2.2 Network-level properties of the max-pressure control

One of the main properties of max-pressure control is its queue length stability, as defined in equation (2.4). The queue length is stable if the expected value of the total queue length is bounded.

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \mathbb{E}\left[\sum_{(i,j)} x_{ij}(t)\right] < \infty \tag{2.4}$$

Some studies proved the stability of the control algorithm analytically under the assumption of infinite queue capacity (Varaiya, 2013; Le et al., 2015; Hsieh et al., 2017) or finite queue capacity (Xiao et al., 2014; Li and Jabari, 2019). Some studies showed the queue length became stable using simulation (Lioris et al., 2016). Existing studies proved the stability of queue length considering only vehicle queues, but Chapter 5 proves the network-level queue length stability considering both pedestrian and vehicle queues.

Due to the challenges in establishing analytical proofs of stability, some studies used only simulations to evaluate performance. Lioris et al. (2016) used simulations to test the efficiency of adaptive max-pressure control on a road section with 16 intersections and 76 links. Queue lengths were shown to be bounded in the simulated examples. Sun and Yin (2018) applied the max-pressure policy of Varaiya (2013), the cyclic phase max-pressure control of Le et al. (2015) and a modified non-cyclic max-pressure model in the simulation. Their simulation results showed that Varaiya's policy resulted in network gridlock. The modified non-cyclic max-pressure policy outperformed the cyclic max-pressure policy. Kouvelas et al. (2014) applied the max-pressure control on an arterial road with 4 intersections. The pressure calculation in their study considered both the link storage capacity and the link queue length. The duration of each phase was calculated by proportionally assigning the available green time based on the weight of each phase.

2.3 Input data for the max-pressure control

Most relevant studies relied on queue length information, but Wu et al. (2017) proposed a control policy using vehicle delays. The weight of turning movement (i, j) was defined as the sojourn time of the head-of-line vehicle in queue (i, j). They claimed that their delay-based control addressed the problem of excessive delay of some queues in queue-based control. The fluid limit model was used to prove the delay-based control was throughput optimal, but the complex nature of vehicle delay and the effects of first-in-first-out behavior resulted in significant limitations on their data inputs. They primarily used the sojourn time of the head-of-line vehicle as a proxy for the queue length, which is hard to collect because the head-of-line vehicle needs to be tracked all the time. Unlike Wu et al.'s study that proposed one form of the input data, Chapter 4 in this thesis requires only mild assumptions for a function to be used in the max-pressure control while maintaining the queue stability of the network. The conclusion can help to determine the feasibility of different data resources to be used in the max-pressure control.

2.4 Autonomous intersection management (AIM)

AIM is a type of signal-free intersection control algorithm. Its application is based on the autonomous driving technology to precisely control vehicles' movements and V2I or V2V technologies to transmit messages between intersection controllers and approaching vehicles. AIM algorithms coordinate vehicles with conflicting trajectories to avoid crashes. As AIM algorithms consider minimizing vehicle delays or vehicle gaps as their objectives and allow vehicles with conflicting trajectories to move in the same time interval, they are more efficient than traditional intersection controls, as highlighted by many studies in their simulations (Fajardo et al., 2011; Li et al., 2013; Kamal et al., 2015; Wu et al., 2015; Levin and Boyles, 2016; Fayazi et al., 2017).

Dresner and Stone (2004) considered an intersection with autonomous vehicles as a multi-agent system and proposed a reservation-based approach to coordinate the reservation of tiles at the intersection for vehicles. Initial studies focused on conflict-free protocols for vehicle trajectory reservations (Dresner and Stone, 2006) with extensions to emergency vehicle and human vehicle access (Dresner and Stone, 2007a,b). AIM was also used to manage inter-connected intersections in a network (Hausknecht et al., 2011) or other types of intersections, such as roundabouts (Bento et al., 2012). The effects of AIM under the context of dynamic traffic assignment was also explored (Zhu and Ukkusuri, 2015).

AIM was formulated with multiple models, such as linear programming (Jin et al., 2012), mixed-integer linear programming (Zhu and Ukkusuri, 2015; Fayazi et al., 2017; Levin and Rey, 2017), mixed-integer nonlinear programming (Mirheli et al., 2019). Other studies formulated this problem in a model predictive control framework (Kamal et al., 2013, 2015) or as a dynamic optimization problem (De Campos et al., 2017; Wuthishuwong and Traechtler, 2013; Mirheli et al., 2018).

Different models of AIM provided different outputs. Some models calculated vehicle arrival times at conflict points or depart times and exit times at the intersection (Jin et al., 2012; Levin and Rey, 2017), while some models gave the number of vehicles allowed to move (Zhu and Ukkusuri, 2015). The objectives of these models included maximizing the total throughput (Fayazi et al., 2017; Levin and Rey, 2017), minimizing total travel times (Jin et al., 2012), minimizing fuel consumption (Zhang et al., 2016), and minimizing potential risk (Kamal et al., 2015).

AIM is originally designed for an intersection with 100% autonomous vehicles so that vehicles can follow the controller's instructions. There are studies that consider AIM with human vehicles that are not equipped or partially equipped with V2V and V2I communication facilities and do not have autonomous driving modules (Dresner and Stone, 2007b; Bento et al., 2013; Qian et al., 2014; Levin and Boyles, 2016). As human drivers are subject to high control uncertainty, these studies considered the vehicle dynamics of human

vehicles and intersections used traffic lights to communicate with human vehicles. Due to the challenges of incorporating human vehicles in AIM, Chapter 5 focuses on the case where all vehicles are autonomous and can communicate with the intersection manager.

Existing studies of AIM optimized vehicle trajectories with all autonomous vehicles or a mix of autonomous and legacy vehicles under various objectives, but none of them consider pedestrians in their model. Chapter 5 proposes an AIM algorithm with pedestrians. Furthermore, Chapter 5 goes beyond adding a simple pedestrian phase to AIM by allowing simultaneous activation of crosswalks and non-conflicting vehicle movements. Adding the crosswalk activation may change the performance of the AIM, which is originally designed for improving vehicle efficiency. For example, the activation of crosswalks may prevent the controller from activating conflicting vehicles.

2.5 Intersection control and user equilibrium

The third topic in this research is about user equilibrium with max-pressure control. In traffic assignment, the travel time and the traffic flow on each link are calculated. User equilibrium is a widely accepted rule for the traffic assignment. Based on Wardrop's first principle, a user equilibrium is achieved when no road user can reduce his or her travel cost through unilaterally changing his or her route. In user equilibrium, all used paths connecting the same pair of origin and destinations have the same travel times. When adaptive intersection controls are applied, the traffic assignment calculation becomes more complex because of the interaction between the intersection control and the route choice behaviors of road users. When road users switch from paths with large travel times to shorter travel times, the traffic assignment changes. Adaptive intersection controls react to the change in the traffic assignment by updating intersection controls, which changes the travel times of road users and leads to further changes in the traffic assignment.

There are some studies on combined traffic assignment and intersection control prob-

lem, but these studies did not focus on max-pressure control. For traditional signal controls, the combined traffic assignment and intersection control problem with traditional intersection controls were explored in many studies. Allsop (1974) was one of the first that designed traffic control when considering the effect of traffic assignment. The calculation of traffic control parameters incorporated the traffic assignment variables. Smith (1979) gave an example in which Webster (1958)'s method did not maximize the travel capacity of a network with three one-way links because changes in route choice reduced the effective network throughput. In existing studies about the combined traffic assignment and intersection control problem, some studies created a bi-level optimization model and solved it by using iterative algorithms (Meneguzzer, 1995; Taale, 2008). Meneguzzer (1995) formulated a network equilibrium problem with asymmetric cost functions which considered both intersection operation and user-optimal route choice. The diagonalization algorithm was applied to solve the model. The algorithm was then applied to a test network of suburban Chicago and the result showed that user equilibrium was achieved in this network. Xiao and Lo (2015) built a day-to-day traffic dynamic model to incorporate both day-to-day route choice adjustments and traffic signal adjustments. They showed the importance of identifying the fixed points for designing intervention strategies. Li et al. (2015) integrated vehicle route guidance with traffic signal optimization. The model was established as a space-phase-time hyper-network. A Lagrangian-relaxation-based optimization framework was used to decouple the problem. The model was tested on medium-size networks and the results showed the high efficiency of the solution algorithm.

Other studies solved the model as a global optimization problem. Chen and Ben-Akiva (1998) formulated the combined problem as a one-level Cournot game, a bi-level Stackelberg game, and a Monopoly game. Gartner and Al-Malik (1996) proposed a combined model for signal control and route choice. The signal control models used flow variables as the input and the traffic assignment used travel times on both links and intersections and considered symmetric intersections between adjacent links. Aziz and Ukkusuri (2012)

presented a mixed-integer program that used a travel time function incorporating the intersection delay and phase lost time. Ukkusuri et al. (2013) proposed a bi-level optimization model that included the network loading, the dynamic user equilibrium, the signal control models. The problem was formulated as a Nash-Cournot game and a Stackelberg game. The algorithm was tested in two small networks. Smith (2015) proposed a pressure-based P_0 policy to calculate the stage green-times that took in vehicle delays at intersection entry lanes. The policy was proven to maximize the capacity in the vertical queuing case and the network was able to converge to equilibrium. Smith and Watling (2016) compared the responsive control policy P_0 with modified max-pressure controls proposed by Varaiya (2013) and Le et al. (2015) by applying three controls on a one-node network. The forms of the original max-pressure control policies in studies of Varaiya and Le et al. were modified to fit in their network model. They showed that both max-pressure controls failed to maximize the throughput but the P_0 policy could.

The study of Smith and Watling (2016) was the only one that explored the user equilibrium in a network with max-pressure control. They used a small network with one intersection to analyze the stability and convergence of the traffic assignment. Besides, their modification to the original max-pressure controls was significantly different from that in studies of Varaiya (2013) and Le et al. (2015) as the weight functions were fully changed. As the weight function plays an important role in the properties of the max-pressure control, modifying it may lead to the wrong conclusion.

Le et al. (2017) is the only one that includes the effect of road users' route choice behavior in the design their max-pressure control policy. Instead of maximizing the intersection total pressure, their control policy aimed to minimize the inequality of traffic in the network, represented by a Gini index function.

2.6 Routing guidance algorithms based on max-pressure control

The fourth topic in this thesis on routing guidance algorithms based on max-pressure control. Max-pressure routing guidance algorithms were originally proposed for reducing delays of data packets of traditional max-pressure algorithms in communication networks. Because the traditional back-pressure algorithm (Tassiulas and Ephremides, 1990) explores all possible routes connecting the origins and destinations, it may send data packets to long routes, which increases travel times and delays, although it has proven queue length stability. The max-pressure routing guidance algorithms are able to guide vehicles to use shorter routes.

There are some proposed max-pressure control algorithms considering routing guidance in communication networks. Ying et al. (2010) proposed a shortest-path-aided max-pressure algorithm that categorized packet flows by the length of routes used by these flows, and route length constraints were considered in the model. Bui et al. (2009) proposed a max-pressure algorithm to reduce delays of the traditional max-pressure control. This study constructed virtual queues for flows with all origin-destination pairs. The serving rate of each link was calculated based on the weight of each flow. In communication networks, delays are mainly generated at servers (nodes) and the travel time between nodes is constant, so studies in communication networks use the number of nodes in a route to approximate the length of the route. However, in traffic networks, delays are produced at nodes and on links. Delays on traffic links result from vehicular interactions, so the travel time of a route can not be simply represented by the number of intersections this path passed through.

There are other studies for traffic networks. Zhang et al. (2012) proposed a max-pressure routing algorithm considering road users' expected travel time. Zaidi et al. (2016) created virtual queues for each traffic flow with the same origin-destination pair and the

virtual queue information was used to calculate the phase activation. In this study, the routing probability was calculated after the optimal phase was determined. Gregoire et al. (2016) proposed a mixed-integer program to calculate the optimal phase and the optimal moving directions of vehicles on each link. The split phasing was used, and all vehicles on the same link will be allowed to move at the same time. In the study of Van Kampen (2015), vehicles' route choices were calculated before the calculation of the optimal phase. To calculate the turning proportion, this study first defined utility functions for selected routes and calculated proportions of each route that connecting the current link to the destination node. In the routing guidance algorithm of Le et al. (2017), a utility function that measured the inequality of traffic within the network was added to the objective function. The pressure function was designed for the split phasing, in which all movements from the same upstream link need to move together.

Among existing studies of routing guidance algorithms based on the max-pressure control, some studies determined the optimal phase and the turning proportion separately (Zaidi et al., 2016; Van Kampen, 2015) because weight functions used in these studies did not incorporate the term of turning proportion, which used the form shown in equation (2.3). For other studies that determined the optimal phase and the turning proportion simultaneously, a phase weight function was used to incorporate the turning proportion term. However, the form of the weight function was only designed for split phasing, which restricted the application of these models to other phase settings.

Chapter 3

Network Model

This chapter introduces the main part of the network model used in the following chapters. Network models used in the following chapters might have some differences but their basic frameworks are the same. The network model in Chapter 5 includes both vehicles and pedestrians, while network models in other chapters only consider vehicles. However, these networks have apparent similarity for vehicles, which will be introduced in this chapter, including the network structure, the model for flow evolution, the intersection control. As the definition of the stability region is highly based on the network model, the stability region in four studies in this research also have similar forms, which is also included in this chapter.

3.1 Network structure

Consider a network $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ including a node set $\mathcal{N}$ and a link set $\mathcal{L}$. The link set is further divided to three subsets: the entry link set $\mathcal{L}_{\text{entry}}$, the internal link set $\mathcal{L}_{\text{int}}$, and the exit link set $\mathcal{L}_{\text{exit}}$. Entry links are source links where vehicles can enter the network and exit links are sinks where vehicles can leave the network. Internal links are links that are either source links nor exit links, which do not connect with any source node or exit node. It is assumed that every link has a free-flow travel time of one time step. A long link can

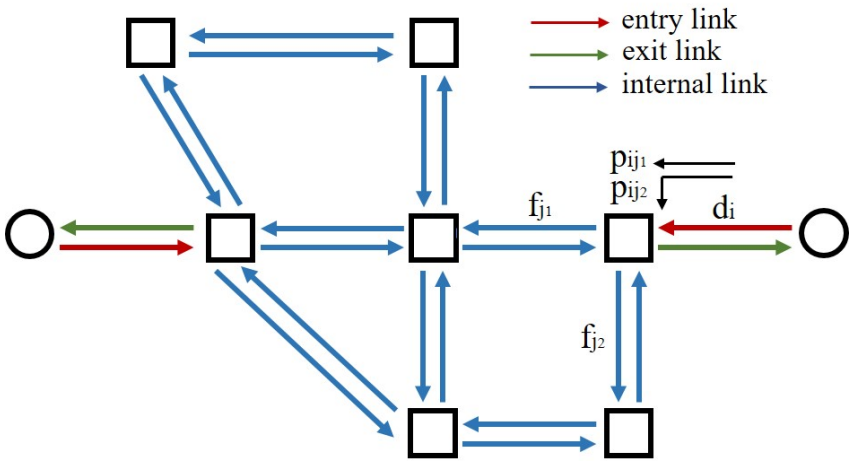


Figure 3.1: An example of the network

be divided into several shorter links whose travel time is the free-flow travel time. Γ_i^+ and Γ_i^- represent the sets of outgoing links and the incoming links of link i respectively. A pair of links represents a movement. For example, (i, j) denotes the movement going from link i to link j. When vehicles in a movement cannot be served in time, the queue will be generated. $\mathcal{M}$ is the set of all movements in the network.

Let $d_i(t)$ be the demand entering the network on link $i \in \mathcal{L}_{\text{entry}}$, and let $\mathbf{d}(t)$ be the vector of demands. For every $i \in \mathcal{L}_{\text{entry}}$, it is assumed that $d_i(t)$ are independent identically-distributed random variables with respect to t. $\bar{\mathbf{d}} = \mathbb{E}[\mathbf{d}(t)]$ is the vector of average demand rates on all entry links. Let $\mathbf{p}(t)$ be the vector of turning proportions for all turning movements at time t. $p_{ij}(t)$ is the percentage of vehicles on link i going to link j at time t. Similarly, it is assumed that $p_{ij}(t)$ are independent identically-distributed random variables. Let $\bar{\mathbf{p}}$ be the vector of average turning proportions of all turning movements, in which $\bar{p}_{ij}$ is the average turning proportion for vehicle on link i to link j. The capacity, which is the maximum service rate of each turning movement, is denoted by Q_{ij}.

3.2 Flow propagation

The store-and-forward model of Varaiya (2013) is used to represent the propagation of traffic flows in the network.

$$x_{ij}(t+1) = x_{ij}(t) - y_{ij}(t) + \sum_{h \in \Gamma_i^-} y_{hi}(t) p_{ij}(t) \qquad \forall i \in \mathcal{L}_{\text{int}}, h \in \Gamma_i^-, j \in \Gamma_i^+ \tag{3.1}$$

$$x_{ij}(t+1) = x_{ij}(t) - y_{ij}(t) + d_i(t) p_{ij}(t) \qquad \forall i \in \mathcal{L}_{\text{entry}}, j \in \Gamma_i^+ \tag{3.2}$$

Equation (3.1) represents the flow propagation for internal links. $x_{ij}(t)$ is the queue length for movement (i, j) at time t. For internal links, queue length $x_{ij}(t+1)$ evolves via conservation of flow. For entry links, the third term in equation (3.1) is replaced by the entering demand $d_i(t)$. As shown in equation (3.3), $y_{ij}(t)$ is the number of vehicles or pedestrians that exit the queue at time t, which is the minimum between the queue length $x_{ij}(t)$ and the capacity Q_{ij} of movement (i, j) when the signal is activated, i.e. $s_{ij}(t) = 1$.

$$y_{ij}(t) = \min \{Q_{ij} s_{ij}(t), x_{ij}(t)\} \tag{3.3}$$

3.3 Expectation of average flow

To evaluate whether a demand vector can be address by an intersection control, the expectation of the average flow $\mathbf{f}$ is calculated first. This is the average flow only when the network is stable, and this average flow cannot be realized when the network is unstable. A vector of average turning proportions of all turning movements $\bar{\mathbf{p}}$ and a vector of average demand rates on all entry links $\bar{\mathbf{d}} = \mathbb{E}[\mathbf{d}(t)]$ can determine the average flow for all links and for all turning movements. As shown in Figure 3.1, once the average demand rate $\bar{d}_i$ on entry link i and the turning proportions of downstream movements $\bar{p}_{ij_1}, \bar{p}_{ij_2}$ are known, the average flows on downstream turning movements f_{ij_1}, f_{ij_2} are known. The relation between the

average flow, the demand rate, and the turning proportion is shown in equations (3.4).

$$f_i = \bar{d}_i \qquad \forall i \in \mathcal{L}_{\text{entry}} \tag{3.4a}$$

$$f_j = \sum_i f_i \bar{p}_{ij} \qquad \forall j \in \mathcal{L}_{\text{int}} \tag{3.4b}$$

3.4 Intersection flow model

This section does not talk about the intersection control algorithm in detail but instead represents the intersection control in the network model. The activation of turning movement (i, j) is denoted by $s_{ij}(t) \in \{0, 1\}$. Let $s_n(t)$ be a matrix denoting the signal control at intersection n. All turning movements activated in intersection control matrix $s_n(t)$ should not conflict with each other. $s(t)$ is a network control matrix, which includes control for all intersections. Let $s = \{s(t), \forall t \in \{1, ..., T\}\}$ denote a network control sequence that includes signal controls for all intersections from the start to the end of the time horizon T. s_n is the intersection control sequence for intersection n. Let $\mathcal{S}$ be a set including all feasible network control matrices for all intersections, while $\mathcal{S}_n$ denotes a set including all feasible intersection control matrices for intersection n. Given an intersection control sequence, the average activation rate of a turning movement can be calculated as follows:

$$\bar{s}_{ij} = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} s_{ij}(t) \tag{3.5}$$

We define the convex hull of all feasible control matrices as $Conv(\mathcal{S})$. $Conv(\mathcal{S})$ is used to connect intersection control sequence with the long term activation rate for all turning movements, which could be fractional numbers between 0 and 1.

We use $\bar{\mathbf{s}}$ to denote a vector of average activation rates of all turning movements in a network. By Proposition 2 of Varaiya (2013), if a network control sequence only uses network control matrices in $\mathcal{S}$, then its corresponding long term activation rate will be in

the convex hull $Conv(\mathcal{S})$.

3.5 Network stability

The network-level queue length stability is a property of max-pressure control. This section provides a framework of proving the stability of the max-pressure control.

The stability of the max-pressure algorithm is related to the network demand $\bar{\mathbf{d}}$ and the average activation rates $\bar{\mathbf{s}}$ corresponding to any network control sequence s. An average demand vector $\bar{\mathbf{d}}$ and an average turning proportion vector $\bar{\mathbf{p}}$ can uniquely determine the flow vector including the average vehicle arrival rate for each turning movement (denoted by $\mathbf{f}$), whose calculation is given in equations (3.4).

The average flow represents the average demand for movement (i, j) or link i be stabilized by some signal timing sequences. Notice that equations (3.4) do not guarantee that flow rates of $\mathbf{f}$ will actually be realized. These flow rates are only achieved when the network is stable. Regardless of the stability, demand for entry links can be converted into demand for turning movements via equations (3.4). If the network is to be stabilized, the demand for each turning movement, $\mathbf{f}$, must be served with adequate capacity.

To make a network control sequence s stabilize a demand vector $\bar{\mathbf{d}}$, the average capacity given to each turning movement should be sufficient to accommodate the average vehicle arrival rate:

$$\bar{s}_{ij} Q_{ij} \geq f_{ij} \tag{3.6}$$

The average activation rate $\bar{s}_{ij}$ of turning movement (i, j) is shown in equation (3.4). A demand vector $\bar{\mathbf{d}}$ is stabilizable if it can be stabilized by a network control sequence S when its corresponding average activation rates $\bar{\mathbf{s}}$ is in convex hull $Conv(\mathcal{S})$. The stable region of demand $\mathcal{D}$ is a set in which every demand vector can be accommodated by some

network control sequences s whose related average activation rate $\bar{\mathbf{s}}$ is in $Conv(\mathcal{S})$. Let $\mathcal{D}^o$ denote the interior of the stable region $\mathcal{D}$. When there is a demand vector $d \in \mathcal{D}^o$, the flow pattern $\mathbf{f}$ corresponding to it satisfies $f_i \bar{p}_{ij} < \bar{s}_{ij} Q_{ij}$, which is also equivalent to $f_i \bar{p}_{ij} + \epsilon = \bar{s}_{ij} Q_{ij}$ for some $\epsilon > 0$.

Definition 1. *The traffic network is strongly stable if the queue length remains bounded in expectation (Neely, 2010):*

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left\{|\mathbf{x}(t)|\right\} < \infty \tag{3.7}$$

Strong stability implies that the queue length of the system under control does not grow without bound in the long run.

Chapter 4

Max-Pressure Control Using Travel Time Functions

Traditional max-pressure control algorithms use the queue length information collected by traffic detectors installed in the field, such as the loop detectors and video cameras. These detectors are expensive to maintain and may have low accuracy or a limited detecting range. Besides, the max-pressure cannot be applied to intersections without these detectors. On-line traffic information from geolocalisation or navigation apps provides another type of data resource that may be applied in the max-pressure control, such as the travel time information. However, it is unknown in what form can the travel time information be applied to max-pressure control. The study introduced in this chapter explores the requirement for a travel time function to be used in max-pressure control to maintain network stability.

4.1 Max-pressure control policy with travel time functions

This chapter uses the same network model introduced in Chapter 3, but the weight function is different from the one mentioned in Section 2.1. Travel time measurements are used in the max-pressure control policy by defining a travel time function to calculate the weight of turning movements. There are several advantages of applying travel times instead of queue lengths to the max-pressure control. It guarantees the fairness of different turning

movements at the intersection. A turning movement with lower queue lengths will not wait as long when the travel time function is used. The data collection of travel times, which can be accomplished by smartphones on vehicles, do not need loop or video detectors which are expensive to install and maintain.

We define $\tau_{ij}(t)$ as a function related to the travel time of turning movement (i, j) at time t. For the purposes of proving stability, we require that function $\tau_{ij}(t)$ is a continuous and monotone increasing function of the queue length. Let $\boldsymbol{\tau}(t)$ be a vector function which includes all values of $\tau_{ij}(t)$ for all turning movements. Intuitively, a long queue length should correspond to a long delay for a vehicle in the queue.

According to Little's law, the average queue length is equal to the average intersection travel time multiplied by the average flow rate, which indicates that intersection travel times and queue lengths are correlated on average. The equation for the weight of a turning movement is given by:

$$w_{ij}(t) = \tau_{ij}(t) - \sum_{k \in \Gamma_j^+} \tau_{jk}(t) p_{jk}(t) \tag{4.1}$$

The equation used to calculate the weight takes the same form as that in Section 2.1, which is the traffic state for the target turning movement minus the weighted average traffic state for the downstream turning movements. The optimal intersection control $s^\star(t)$ is calculated by equation (1.2).

4.2 Network stability with travel time function

In this section, the assumption for a travel time function to maintain the network queue-length stability is proposed. The queue length stability is defined in equation (3.7). The evolution of vehicle queues can be described by a discrete time Markov chain with the state vector $\mathbf{x}(t)$. Let $v(\mathbf{x}(t)) \geq 0$ represent a Lyapunov function. The one-step conditional

Lyapunov drift can be defined as follows:

$$\Delta(\mathbf{x}(t)) \triangleq \mathbb{E}\left[v(\mathbf{x}(t+1)) - v(\mathbf{x}(t))|\mathbf{x}(t)\right] \tag{4.2}$$

Lemma 1. *For a given Lyapunov function $v(\mathbf{x}(t))$, assume $v(\mathbf{x}(0))$ is a finite value. If there exists a constants $B < \infty$ and $\xi > 0$ such that the Lyapunov drift satisfies equation (4.3) for all time interval t and all possible values of $\mathbf{x}(t)$, then the traffic network is strongly stable.*

$$\Delta(\mathbf{x}(t)) \leq B - \xi|\mathbf{x}(t)| \tag{4.3}$$

Proof. We could add up both sides of the inequality (4.3) for all time steps from $t = 0$ to $t = T - 1$ and taking expectations, then we can get:

$$\begin{aligned}\sum_{t=0}^{T-1}\Delta(\mathbf{x}(t)) &= \sum_{t=0}^{T-1}\left(\mathbb{E}\left[v(\mathbf{x}(t+1))|\mathbf{x}(t)\right] - \mathbb{E}\left[v(\mathbf{x}(t))|\mathbf{x}(t)\right]\right) \\ &= \mathbb{E}\left[v(\mathbf{x}(T))\right] - \mathbb{E}\left[v(\mathbf{x}(0))\right] \leq TB - \epsilon\sum_{t=0}^{T-1}\mathbb{E}\left[|\mathbf{x}(t)|\right]\end{aligned} \tag{4.4}$$

Then we can get:

$$\frac{1}{T}\sum_{t=0}^{T-1}|\mathbf{x}(t)| \leq \frac{B}{\epsilon} + \frac{1}{\epsilon T}\mathbb{E}\left[v(\mathbf{x}(0))\right] - \frac{1}{\epsilon T}\mathbb{E}\left[v(\mathbf{x}(T))\right] \leq \frac{B}{\epsilon} + \frac{1}{\epsilon T}\mathbb{E}\left[v(\mathbf{x}(0))\right] \tag{4.5}$$

When taking the limit of $t \to \infty$ on both sides, the second term on the right hand side becomes zero, then we can get:

$$\limsup_{t\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}|\mathbf{x}(t)| \leq \frac{B}{\epsilon} \tag{4.6}$$

□

Based on Lemma 1, when the state has a large value, the Lyapunov drift will be a negative value and the Lyapunov function will decrease. In this thesis, the Lyapunov function is defined as $v(\mathbf{x}(t)) = \sum_{(i,j)\in\mathcal{M}} \tau_{ij}(t)x_{ij}(t)$.

we decompose the Lyapunov drift as follows:

$$\begin{aligned}
&\Delta(\mathbf{x}(t)) = \mathbb{E}\left[v(t+1) - v(t)|\mathbf{x}(t)\right] \\
&= \mathbb{E}\left[\sum_{(i,j)\in\mathcal{M}} (\tau_{ij}(t+1) - \tau_{ij}(t))x_{ij}(t)|\mathbf{x}(t)\right] \\
&+ \mathbb{E}\left[\sum_{(i,j)\in\mathcal{M}} (x_{ij}(t+1) - x_{ij}(t))\tau_{ij}(t)|\mathbf{x}(t)\right] \\
&+ \mathbb{E}\left[\sum_{(i,j)\in\mathcal{M}} (\tau_{ij}(t+1) - \tau_{ij}(t))\,(x_{ij}(t+1) - x_{ij}(t))|\mathbf{x}(t)\right] \\
&= \alpha_1 + \alpha_2 + \alpha_3 \quad (4.7)
\end{aligned}$$

α_1, α_2, and α_3 are proved to be bounded or followed the form in equation (4.3) using the properties of function $\tau_{ij}(t)$. The uppers bounds or simplified form of α_1, α_2, and α_3 are includes in Lemmas 2, 3, and 4, based on which Proposition 1 is proposed.

Lemma 2.

$$\alpha_3 = \mathbb{E}\left[\sum_{(i,j)\in\mathcal{M}} (\tau_{ij}(t+1) - \tau_{ij}(t))\,(x_{ij}(t+1) - x_{ij}(t))|\mathbf{x}(t)\right] \leq K_1K_2|\mathcal{M}|$$

Proof. α_3 is transformed as follows:

$$\mathbb{E}\left[\sum_{(i,j)\in\mathcal{M}} (\tau_{ij}(t+1) - \tau_{ij}(t))\,(x_{ij}(t+1) - x_{ij}(t))|\mathbf{x}(t)\right]$$

$$= \mathbb{E}\left[\sum_{(i,j)\in\mathcal{L}_{\text{int}}\times\mathcal{L}} (\tau_{ij}(t+1)-\tau_{ij}(t))\left(-y_{ij}(t)+\sum_{h\in\Sigma_i^-} y_{hi}p_{ij}(t)\right)|\mathbf{x}(t)\right]$$

$$+\mathbb{E}\left[\sum_{(i,j)\in\mathcal{L}_{\text{entry}}\times\mathcal{L}} (\tau_{ij}(t+1)-\tau_{ij}(t))\left(-y_{ij}(t)+d_{ij}(t)\right)|\mathbf{x}(t)\right] \quad (4.8)$$

Let $\hat{Q}_{ij}$ denote this upper bound for the capacity of turning movement (i,j) and let $\hat{d} = \arg\max_{ij}\{d_{ij}(t)\}$ denote the largest demand rate going to an entry link.

$$\left(-y_{ij}(t)+\sum_{h\in\Sigma_i^-} y_{hi}p_{ij}(t)\right) \le \max\left\{\hat{Q}_{ij}, \sum_{h\in\Sigma_i^-}\hat{Q}_{hi}\right\}, \forall i \in \mathcal{L}_{\text{int}} \quad (4.9)$$

$$(-y_{ij}(t)+d_{ij}(t)) \le \max\left\{\hat{Q}_{ij}, \hat{d}_{ij}\right\}, \forall i \in \mathcal{L}_{\text{entry}} \quad (4.10)$$

Therefore, $(x_{ij}(t+1)-x_{ij}(t)) \le \max\left\{\hat{Q}_{ij}, \sum_{h\in\Gamma_i^-}\hat{Q}_{hi}, \hat{d}_{ij}\right\} = K_1$. To find a bound for the term $(\tau_{ij}(t+1)-\tau_{ij}(t))$, it is necessary to relate function $\tau_{ij}(t)$ with $x_{ij}(t)$ as $x_{ij}(t+1)-x_{ij}(t)$ is bounded by the term $\max\left\{\hat{Q}_{ij}, \sum_{h\in\Gamma_i^-}\hat{Q}_{hi}, \hat{d}_{ij}\right\}$. As we assume function $\tau_{ij}(t)$ is a continuous and strictly increasing function of $x_{ij}(t)$, $(\tau_{ij}(t+1)-\tau_{ij}(t))$ is also bounded. Let K_2 to denote the upper bound. Let $|\mathcal{M}|$ denote the total number of turning movements. Therefore,

$$\alpha_3 = \mathbb{E}\left[\sum_{(i,j)\in\mathcal{M}} (\tau_{ij}(t+1)-\tau_{ij}(t))\,(x_{ij}(t+1)-x_{ij}(t))|\mathbf{x}(t)\right] \le K_1K_2|\mathcal{M}|$$

□

Lemma 3.

$$\alpha_2 = \mathbb{E}\left[\sum_{(i,j)\in\mathcal{M}} (x_{ij}(t+1)-x_{ij}(t))\tau_{ij}(t)|\mathbf{x}(t)\right] \le \sum_{(i,j)\in\mathcal{M}} Q_{ij}^2 - \epsilon\eta|\boldsymbol{\tau}(t)|$$

Proof. At first, α_2 is transformed as follows:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{(i,j)\in\mathcal{M}}(x_{ij}(t+1)-x_{ij}(t))\tau_{ij}(t)|\mathbf{x}(t)\right] \\
&=\mathbb{E}\left[\sum_{(i,j)\in\mathcal{L}_{\text{int}}\times\mathcal{L}}\left(-y_{ij}(t)+\sum_{h\in\Sigma_i^-}y_{hi}p_{ij}(t)\right)\tau_{ij}(t)|\mathbf{x}(t)\right] \\
&\qquad\qquad+\mathbb{E}\left[\sum_{(i,j)\in\mathcal{L}_{\text{entry}}\times\mathcal{L}}\left(-y_{ij}(t)+d_{ij}(t)\right)\tau_{ij}(t)|\mathbf{x}(t)\right] \\
&=\mathbb{E}\left[\sum_{(i,j)\in\mathcal{M}}y_{ij}(t)\left(-\tau_{ij}(t)+\sum_{k\in\Sigma_j^+}p_{jk}(t)\tau_{jk}(t)\right)|\mathbf{x}(t)\right] \\
&\qquad\qquad+\mathbb{E}\left[\sum_{(i,j)\in\mathcal{L}_{\text{entry}}\times\mathcal{L}}d_{ij}(t)\tau_{ij}(t)|\mathbf{x}(t)\right]
\end{aligned}$$

Since $w_{ij}(t)=\tau_{ij}(t)-\sum_{k\in\Sigma_j^+}\tau_{jk}(t)p_{jk}(t)$, we obtain:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{(i,j)\in\mathcal{M}}(x_{ij}(t+1)-x_{ij}(t))\tau_{ij}(t)|\mathbf{x}(t)\right]= \\
&\qquad\mathbb{E}\left[\sum_{(i,j)\in\mathcal{M}}y_{ij}(t)(-w_{ij}(t))|\mathbf{x}(t)\right]+\sum_{(i,j)\in\mathcal{L}_{\text{entry}}\times\mathcal{L}}\bar{d}_{ij}\tau_{ij}(t) \quad (4.11)
\end{aligned}$$

The second term of equation (4.11) can be transformed to obtain weight $w_{ij}(t)$.

$$\begin{aligned}
\sum_{(i,j)\in\mathcal{L}_{\text{entry}}\times\mathcal{L}}\bar{d}_{ij}\tau_{ij}(t)&=\sum_{(i,j)\in\mathcal{L}_{\text{entry}}\times\mathcal{L}}\bar{d}_i\bar{p}_{ij}\tau_{ij}(t)=\sum_{(i,j)\in\mathcal{M}}f_i\bar{p}_{ij}\tau_{ij}(t)-\sum_{(j,k)\in\mathcal{L}_{\text{int}}\times\mathcal{L}}f_j\bar{p}_{jk}\tau_{jk}(t) \\
&=\sum_{(i,j)\in\mathcal{M}}f_i\bar{p}_{ij}\tau_{ij}(t)-\sum_{(j,k)\in\mathcal{L}_{\text{int}}\times\mathcal{L}}\left(\sum_{(i,j)\in\mathcal{M}}f_i\bar{p}_{ij}\right)\bar{p}_{jk}\tau_{jk}(t)
\end{aligned}$$

$$= \sum_{(i,j)\in\mathcal{M}} f_i \bar{p}_{ij} w_{ij}(t) \quad (4.12)$$

After rewriting the second term in equation (4.11), we can combine two terms in equation (4.11), after which the intersection control can be included.

$$\sum_{(i,j)\in\mathcal{M}} f_i \bar{p}_{ij} w_{ij}(t) - \sum_{(i,j)\in\mathcal{M}} \mathbb{E}\left[y_{ij}(t)|\mathbf{x}(t)\right] w_{ij}(t)$$
$$= \sum_{(i,j)\in\mathcal{M}} \left(f_{ij} - \mathbb{E}\left[y_{ij}(t)|\mathbf{x}(t)\right]\right) w_{ij}(t) \quad (4.13)$$

As the demand vector $\bar{\mathbf{d}}$ is stabilizable, i.e. $\bar{\mathbf{d}} \in \mathcal{D}^o$, $\bar{s}_{ij} Q_{ij} > f_{ij}$ from equation (3.6). We introduce $s_{ij}(t)Q_{ij}$ in equation (4.13) to obtain equation (4.14).

$$\sum_{(i,j)\in\mathcal{M}} \left(f_{ij} - \mathbb{E}\left[y_{ij}(t)|\mathbf{x}(t)\right]\right) w_{ij}(t) =$$
$$\sum_{(i,j)\in\mathcal{M}} \left[\left(f_{ij} - s_{ij}(t)Q_{ij}\right) + \left(s_{ij}(t)Q_{ij} - \mathbb{E}\left[y_{ij}(t)|\mathbf{x}(t)\right]\right)\right] w_{ij}(t) \quad (4.14)$$

Inequalities $s_{ij}(t)Q_{ij} \geq \mathbb{E}\left[y_{ij}(t)|\mathbf{x}(t)\right]$ and $w_{ij}(t) \leq x_{ij}(t)$ always hold. The relation between $s_{ij}(t)Q_{ij}$ and $y_{ij}(t)$ is affected by the value of $x_{ij}(t)$, as shown in equation (4.15).

$$\left(s_{ij}(t)Q_{ij} - \mathbb{E}\left[y_{ij}(t)|\mathbf{x}(t)\right]\right) w_{ij}(t)$$
$$= \begin{cases} 0, & \text{if } x_{ij}(t) \geq Q_{ij} \\ \left(s_{ij}(t)Q_{ij} - \mathbb{E}\left[y_{ij}(t)|\mathbf{x}(t)\right]\right) w_{ij}(t), & \text{otherwise} \end{cases} \quad (4.15)$$

Therefore, $\sum_{(i,j)\in\mathcal{M}} \left(s_{ij}(t)Q_{ij} - \mathbb{E}\left[y_{ij}(t)|\mathbf{x}(t)\right]\right) w_{ij}(t)$ (when $x_{ij}(t) < Q_{ij}$) is bounded:

$$\sum_{(i,j)\in\mathcal{M}} \left(s_{ij}(t)Q_{ij} - \mathbb{E}\left[y_{ij}(t)|\mathbf{x}(t)\right]\right) w_{ij}(t)$$

$$\leq \sum_{(i,j)\in\mathcal{M}} \left(s_{ij}(t)Q_{ij} - \mathbb{E}\left[y_{ij}(t)|\mathbf{x}(t)\right]\right) x_{ij}(t) \quad (4.16)$$

When $x_{ij}(t) \geq Q_{ij}$, the term in equation (4.15) is zero. When $x_{ij}(t) < Q_{ij}$, the term is a positive value. Therefore:

$$\sum_{(i,j)\in\mathcal{M}} \left(s_{ij}(t)Q_{ij} - \mathbb{E}\left[y_{ij}(t)|\mathbf{x}(t)\right]\right) x_{ij}(t) \leq \sum_{(i,j)\in\mathcal{M}} Q_{ij}^2 \quad (4.17)$$

The term $\left(f_{ij} - s_{ij}(t)Q_{ij}\right) w_{ij}(t)$ is affected by the intersection control policy. If the max-pressure control is used, the phase with largest pressure is activated at every time step. As the demand is feasible, there should be a stationary control $\bar{s}$ to stabilize the network, and the relation between the average service rate and the average flow rate should be

$$\bar{s}_{ij}Q_{ij} = \begin{cases} f_{ij} + \epsilon, & \text{if } w_{ij} > 0 \\ 0, & \text{otherwise} \end{cases} \quad (4.18)$$

Let $s_{ij}^{\star}(t)$ be the calculated optimal control for turning movement (i,j) at time t, which is included in the optimal control $\mathbf{s}^{\star}(t)$. Then we have:

$$\begin{aligned} &\sum_{(i,j)\in\mathcal{M}} \left[f_i p_{ij} - s_{ij}^{\star}(t)Q_{ij}\right] w_{ij}(t) \leq \sum_{(i,j)\in\mathcal{M}} \left[f_i p_{ij} - \bar{s}_{ij}(t)Q_{ij}\right] w_{ij} \\ &= -\epsilon \sum_{(i,j)\in\mathcal{M}, w_{ij}\geq 0} w_{ij}(t) + \sum_{(i,j)\in\mathcal{M}, w_{ij}<0} f_i p_{ij} w_{ij}(t) \\ &= -\epsilon \sum_{(i,j)\in\mathcal{M}, w_{ij}>0} w_{ij}(t) - \sum_{(i,j)\in\mathcal{M}, w_{ij}<0} f_i p_{ij} |w_{ij}(t)| \\ &\leq -\epsilon \sum_{i\in\mathcal{L}_{\text{int}}\cup\mathcal{L}_{\text{entry}}} \max\left(0, w_{ij}(t)\right) \leq -\epsilon|\mathbf{w}(t)| \end{aligned} \quad (4.19)$$

Because the weight is a linear function of $\boldsymbol{\tau}(t)$, then we can find $\eta > 0$ that satisfies:

$$|\mathbf{w}(t)| \geq \eta |\boldsymbol{\tau}(t)|.$$

Then equation (4.19) becomes:

$$\sum_{(i,j)\in\mathcal{M}} \left[f_i p_{ij} - s_{ij}(t) Q_{ij} \right] w_{ij}(t) \leq -\epsilon\eta |\boldsymbol{\tau}(t)| \tag{4.20}$$

Then, the upper bound for α_2 is derived as in equation (4.21).

$$\alpha_2 = \mathbb{E}\left[\sum_{(i,j)\in\mathcal{M}} (x_{ij}(t+1) - x_{ij}(t)) \tau_{ij}(t) | \mathbf{x}(t) \right] \leq \sum_{(i,j)\in\mathcal{M}} Q_{ij}^2 - \epsilon\eta |\boldsymbol{\tau}(t)| \tag{4.21}$$

□

The form of the term $\sum_{(i,j)\in\mathcal{M}} Q_{ij}^2 - \epsilon\eta|\boldsymbol{\tau}(t)|$ is similar to the right hand side of equation (4.3) for the Lyapunov drift. The only difference is that the term $\sum_{(i,j)\in\mathcal{M}} Q_{ij}^2 - \epsilon\eta|\boldsymbol{\tau}(t)|$ is related to function $\boldsymbol{\tau}$ instead of x. However, if $\boldsymbol{\tau}$ is a function of x and is monotone increasing with x, then $\sum_{(i,j)\in\mathcal{M}} Q_{ij}^2 - \epsilon\eta|\boldsymbol{\tau}(t)|$ should have the same form as $B - \epsilon|\mathbf{x}(t)|$.

Lemma 4.

$$\alpha_1 = \mathbb{E}\left[\sum_{(i,j)\in\mathcal{M}} (\tau_{ij}(t+1) - \tau_{ij}(t)) x_{ij}(t) | \mathbf{x}(t), s^\star(t) \right] \leq K_3$$

Proof. $s^\star(t)$ is a vector including all optimal controls for turning movements. The expected value of the queue length difference of two time steps is derived conditional on the average service rate $\bar{\mathbf{s}}$. The average demand vector $\bar{\mathbf{d}}$ is feasible so the average flow rate f_{ij} satisfy the relation $f_{ij} < \bar{s}_{ij} Q_{ij}$ for stationary control $\bar{s}_{ij}$ in $Conv(\mathcal{S})$. $\bar{s}$ is the stationary control. At first, we look at the expected value of the difference in the queue length. As the average

service rate is larger than the average demand rate, the expected value of queue lengths between two time intervals should decrease, which is shown in equation (4.22).

$$\mathbb{E}\left[x_{ij}(t+1) - x_{ij}(t)|\mathbf{x}(t), \bar{s}\right] \leq 0 \tag{4.22}$$

Assuming $\boldsymbol{\tau}$ is an increasing function of x, then the control sequence $\bar{s}$ can reduce the average value of function $\boldsymbol{\tau}$.

$$\mathbb{E}\left[\tau_{ij}(t+1) - \tau_{ij}(t)|\mathbf{x}(t)\right] \leq 0 \tag{4.23}$$

In max-pressure control, the value of the objective term using the optimal control sequence $s^\star$ should be larger than any other control sequence $\bar{s}$ that can address the demand.

$$\sum_{(i,j)\in\mathcal{M}} s^\star_{ij} Q_{ij} \left(\tau_{ij}(t) - \sum_{k\in\Gamma_j^+} \bar{p}_{jk}\tau_{jk}(t)\right) \geq \sum_{(i,j)\in\mathcal{M}} \bar{s}_{ij} Q_{ij} \left(\tau_{ij}(t) - \sum_{k\in\Gamma_j^+} \bar{p}_{jk}\tau_{jk}(t)\right) \tag{4.24}$$

The term $s_{ij}(t)Q_{ij}$ in $\sum_{ij} s_{ij}(t)Q_{ij}\left(\tau_{ij}(t) - \sum_{k\in\Gamma_j^+} p_{jk}(t)\tau_{jk}(t)\right)$ can be replaced by $y_{ij}(t)$ and the difference between two terms $\sum_{ij} s_{ij}(t)Q_{ij}\left(\tau_{ij}(t) - \sum_{k\in\Gamma_j^+} p_{jk}(t)\tau_{jk}(t)\right)$ and $\sum_{ij} y_{ij}(t)\left(\tau_{ij}(t) - \sum_{k\in\Gamma_j^+} p_{jk}(t)\tau_{jk}(t)\right)$ can be proved to be bounded.

As $y_{ij}(t) = \min\left(x_{ij}(t), s_{ij}(t)Q_{ij}\right)$ and $s_{ij}(t)Q_{ij} - y_{ij}(t) \geq 0$, we get:

$$s_{ij}(t)Q_{ij} - y_{ij}(t) = s_{ij}(t)Q_{ij} - \min\left(x_{ij}(t), s_{ij}(t)Q_{ij}\right) \geq 0 \tag{4.25}$$

The activation of turning movement s_{ij} is highly related to its weight. If weight $w_{ij}(t) < 0$, $s_{ij} = 0$. If weight $w_{ij}(t) \geq 0$, $s_{ij} \in \{0, 1\}$.

Therefore, we can get a range for $\sum_{ij} (s_{ij}(t)Q_{ij} - y_{ij}(t)) \left(\tau_{ij}(t) - \sum_{k \in \Gamma_j^+} p_{jk}(t)\tau_{jk}(t) \right)$, which is shown in equation (4.26). The lower bound for the difference is 0.

$$\sum_{(i,j)\in\mathcal{M}} (s_{ij}(t)Q_{ij} - y_{ij}(t)) \left(\tau_{ij}(t) - \sum_{k \in \Gamma_j^+} p_{jk}(t)\tau_{jk}(t) \right) \begin{cases} = 0, & \text{if } w_{ij}(t) < 0 \vee y_{ij}(t) = s_{ij}(t)Q_{ij} \\ \geq 0, & \text{otherwise} \end{cases} \tag{4.26}$$

The term $s_{ij}(t)Q_{ij}$ is less than Q_{ij} and the term $\tau_{ij}(t) - \sum_{k \in \Gamma_j^+} p_{jk}(t)\tau_{jk}(t)$ is less than $\tau_{ij}(t)$. As $\boldsymbol{\tau}(t)$ is a monotone increasing function of $x(t)$, we can find a constant γ to satisfy $\gamma|\boldsymbol{\tau}(t)| \leq |\mathbf{x}(t)|$. When $x_{ij}(t) \leq Q_{ij}$, $s_{ij}(t)Q_{ij} - y_{ij}(t) = 0$, therefore the term $\sum_{ij} (s_{ij}(t)Q_{ij} - y_{ij}(t)) \left(\tau_{ij}(t) - \sum_{k \in \Gamma_j^+} p_{jk}(t)\tau_{jk}(t) \right)$ only collects the values when $\gamma\tau_{ij}(t) \leq x_{ij}(t) \leq Q_{ij}$. As $\tau_{ij}(t) \leq \frac{Q_{ij}}{\gamma}$, we can get the upper bound as shown in equation (4.27).

$$\sum_{(i,j)\in\mathcal{M}} (s_{ij}(t)Q_{ij} - y_{ij}(t)) \left(\tau_{ij}(t) - \sum_{k \in \Gamma_j^+} p_{jk}(t)\tau_{jk}(t) \right) \leq \sum_{ij} \frac{Q_{ij}^2}{\gamma} = K_3 \tag{4.27}$$

When $s_{ij}(t)Q_{ij} \geq y_{ij}(t)$, the queue $x_{ij}(t)$ is larger than the capacity Q_{ij}. If function $\boldsymbol{\tau}$ is monotone increasing with x, the maximum value of $\tau_{ij}(t)$ can be calculated as $\tau_{ij}(Q_{ij})$ given the maximum value for $x_{ij}(t)$, which is Q_{ij}.

We can rewrite $\sum_{ij} s_{ij}(t)Q_{ij} \left(\tau_{ij}(t) - \sum_{k \in \Gamma_j^+} p_{jk}(t)\tau_{jk}(t) \right)$ as follows:

$$\sum_{(i,j)\in\mathcal{M}} s_{ij}(t)Q_{ij}\left(\tau_{ij}(t)-\sum_{k\in\Gamma_j^+} p_{jk}(t)\tau_{jk}(t)\right)= \sum_{jk}\tau_{jk}(t)\left(s_{jk}(t)Q_{jk}-p_{jk}\sum_{i\in\Gamma_j^-} s_{ij}(t)Q_{ij}\right) \quad (4.28)$$

The range for the term $\sum_{jk}\tau_{jk}(t)\left(s_{jk}(t)Q_{jk}-p_{jk}\sum_{i\in\Gamma_j^-} s_{ij}(t)Q_{ij}\right)$ can be derived in equations (4.29).

$$\sum_{jk}\tau_{jk}(t)\left(y_{jk}(t)-p_{jk}\sum_{i\in\Gamma_j^-} y_{ij}(t)\right)\le\sum_{jk}\tau_{jk}(t)\left(s_{jk}(t)Q_{jk}-p_{jk}\sum_{i\in\Gamma_j^-} s_{ij}(t)Q_{ij}\right) \le\sum_{jk}\tau_{jk}(t)\left(y_{jk}(t)-p_{jk}\sum_{i\in\Gamma_j^-} y_{ij}(t)\right)+K_3 \quad (4.29)$$

If the optimal control sequence s^* is inserted in equations (4.29), we can get the relation between $s_{ij}^*Q_{ij}$ and $y_{ij}^\star$ in equations (4.30) , where $y_{ij}^\star$ is the flow when the optimal control sequence is applied.

$$\sum_{jk}\tau_{jk}(t)\left(y_{jk}^\star(t)-p_{jk}\sum_{i\in\Gamma_j^-} y_{ij}^\star(t)\right)\le\sum_{jk}\tau_{jk}(t)\left(s_{jk}^\star(t)Q_{jk}-p_{jk}\sum_{i\in\Gamma_j^-} s_{ij}^\star(t)Q_{ij}\right) \le\sum_{jk}\tau_{jk}(t)\left(y_{jk}^\star(t)-p_{jk}\sum_{i\in\Gamma_j^-} y_{ij}^\star(t)\right)+K_3 \quad (4.30)$$

Similarly, for the stationary control $\bar{s}$ that can address the demand, we can get a similar relation, as shown in equations (4.31).

$$\sum_{jk} \tau_{jk}(t) \left(\bar{y}_{jk}(t) - p_{jk} \sum_{i \in \Gamma_j^-} \bar{y}_{ij}(t) \right) \leq \sum_{jk} \tau_{jk}(t) \left(\bar{s}_{jk}(t) Q_{jk} - p_{jk} \sum_{i \in \Gamma_j^-} \bar{s}_{ij}(t) Q_{ij} \right)$$
$$\leq \sum_{jk} \tau_{jk}(t) \left(\bar{y}_{jk}(t) - p_{jk} \sum_{i \in \Gamma_j^-} \bar{y}_{ij}(t) \right) + K_3 \quad (4.31)$$

We get an upper bound for $-\sum_{jk} \tau_{jk}(t) \left(y^\star_{jk}(t) - p_{jk} \sum_i y^\star_{ij}(t) \right)$ consider the relation between $s^\star$ and $\bar{s}$, as shown in equation (4.32).

$$-\sum_{jk} \tau_{jk}(t) \left(y^\star_{jk}(t) - p_{jk} \sum_i y^\star_{ij}(t) \right)$$
$$\leq -\sum_{jk} \tau_{jk}(t) \left(s^\star_{jk}(t) Q_{jk} - p_{jk} \sum_i s^\star_{ij}(t) Q_{ij} \right) + K_3$$
$$\leq -\sum_{jk} \tau_{jk}(t) \left(\bar{s}_{jk}(t) Q_{jk} - p_{jk} \sum_i \bar{s}_{ij}(t) Q_{ij} \right) + K_3$$
$$\leq -\sum_{jk} \tau_{jk}(t) \left(\bar{y}_{jk}(t) - p_{jk} \sum_i \bar{y}_{ij}(t) \right) + K_3 \quad (4.32)$$

Therefore, we get the relation between $y^\star$ and $\bar{y}$:

$$-\sum_{jk} \tau_{jk}(t) \left(y^\star_{jk}(t) - p_{jk}(t) \sum_i y^\star_{ij}(t) \right)$$
$$\leq -\sum_{jk} \tau_{jk}(t) \left(\bar{y}_{ij} - p_{jk}(t) \sum_i \bar{y}_{ij}(t) \right) + K_3 \quad (4.33)$$

Since $x_{jk}(t+1) - x_{jk}(t) = -y_{jk}(t) + p_{jk}(t) \sum_i y_{ij}(t)$,

$$\mathbb{E}\left[\sum_{jk} \tau_{jk}(t) \left(x_{jk}(t+1) - x_{jk}(t) \right) | s^\star(t) \right] \leq$$

$$\mathbb{E}\left[\sum_{jk}\tau_{jk}(t)\left(x_{jk}(t+1)-x_{jk}(t)\right)|\bar{s}\right]+K_3 \quad (4.34)$$

Since $\tau_{jk}(t)x_{jk}(t)$ is fixed at time step t (with respect to $(t+1)$),

$$\mathbb{E}\left[\sum_{jk}\tau_{jk}(t)x_{jk}(t+1)|s^{\star}(t)\right]\leq\mathbb{E}\left[\sum_{jk}\tau_{jk}(t)x_{jk}(t+1)|\bar{s}\right]+K_3 \quad (4.35)$$

Because we assume that $\tau_{jk}(t)$ is monotone increasing with respect to $x_{jk}(t)$, then:

$$\mathbb{E}\left[\sum_{jk}\tau_{jk}(t+1)x_{jk}(t)|s^{\star}(t)\right]\leq\mathbb{E}\left[\sum_{jk}\tau_{jk}(t+1)x_{jk}(t)|\bar{s}\right]+K_3 \quad (4.36)$$

Then:

$$\mathbb{E}\left[\sum_{jk}\left(\tau_{jk}(t+1)-\tau_{jk}(t)\right)x_{jk}(t)|s^{\star}(t)\right]\leq \mathbb{E}\left[\sum_{jk}\left(\tau_{jk}(t+1)-\tau_{jk}(t)\right)x_{jk}(t)|\bar{s}\right]+K_3 \quad (4.37)$$

By equation (4.23), $\mathbb{E}\left[\sum_{jk}\left(\tau_{jk}(t+1)-\tau_{jk}(t)\right)x_{jk}(t)|\bar{s}\right]\leq 0$, so:

$$\mathbb{E}\left[\sum_{jk}\left(\tau_{jk}(t+1)-\tau_{jk}(t)\right)x_{jk}(t)|s^{\star}(t)\right]\leq K_3 \quad (4.38)$$

□

Proposition 1. *if* $\bar{\mathbf{d}}$ *in* D^o*, the queue length is stable under the max-pressure control.*

Proof. Based on Lemmas 2, 3, and 4, we get:

$$\alpha_1 = \mathbb{E}\left[\sum_{(i,j)\in\mathcal{M}} (\tau_{ij}(t+1) - \tau_{ij}(t))x_{ij}(t)|s^\star\right] \leq K_3 \tag{4.39}$$

$$\alpha_2 = \mathbb{E}\left[\sum_{(i,j)\in\mathcal{M}} (x_{ij}(t+1) - x_{ij}(t))\tau_{ij}(t)\right] \leq \sum_{(i,j)\in\mathcal{M}} Q_{ij}^2 - \epsilon\eta|\boldsymbol{\tau}(t)| \tag{4.40}$$

$$\alpha_3 = \sum_{(i,j)\in\mathcal{M}} (\tau_{ij}(t+1) - \tau_{ij}(t))\,(x_{ij}(t+1) - x_{ij}(t)) \leq K_1K_2|\mathcal{M}| \tag{4.41}$$

Let $\mathcal{K} = K_1K_2|\mathcal{M}| + \sum_{(i,j)\in\mathcal{M}} Q_{ij}^2 + K_3$, we can use this constant to obtain an upper bound on the Lyapunov drift, as shown in equation (4.42).

$$\mathbb{E}\left[v(t+1) - v(t)|\mathbf{x}(t)\right] \leq \mathcal{K} - \epsilon\eta|\boldsymbol{\tau}(t)|, \;\; \forall t \in \{1, 2, ..., T\} \tag{4.42}$$

According to the definition of strongly stability in equation (4.3), we can set $B = \mathcal{K}$, and $\xi = \epsilon\eta$, which yields:

$$\mathbb{E}\left[v(t+1) - v(t)|\mathbf{x}(t)\right] \leq B - \xi|\boldsymbol{\tau}(t)| \tag{4.43}$$

Because we assume $\tau_{ij}(t)$ is monotone increasing with respect to the queue length $x_{ij}(t)$, we can find a positive coefficient λ which satisfies $\boldsymbol{\tau}(t) \geq \lambda\mathbf{x}(t)$. Then equation (4.43) can be transformed into equation (4.44). According to Lemma 1, the stability can be proven.

$$\mathbb{E}\left[v(t+1) - v(t)|\mathbf{x}(t)\right] \leq B - \xi|\boldsymbol{\tau}(t)| \leq B - \xi\lambda|\mathbf{x}(t)| \tag{4.44}$$

□

4.3 Properties of travel time function τ

In Section 4.2, we proved the stability of the queue length when the travel time function $\tau_{ij}(t)$ is a continuous and monotone increasing function of $x_{ij}(t)$. Equivalently, this means that when the average serving rate is larger, the queue length decreases, the travel time decreases. The travel time function $\tau_{ij}(t)$ should also be defined in a way that is similar to the travel times available from explicit data. Equation (4.45) shows one form of the travel time function that satisfy all assumptions and is also related to the travel times.

$$\tau_{ij}(t) = \frac{x_{ij}(t)}{Q_{ij}(t)} \tag{4.45}$$

In equation (4.45), Q_{ij} represents the maximum service rate of turning movement (i, j) and $x_{ij}(t)$ is the queue length, so $\frac{x_{ij}(t)}{Q_{ij}(t)}$ can be considered as an approximation of the time required to empty the queue. As $\tau_{ij}(t)$ is a monotone increasing function with respect to $x_{ij}(t)$, the max-pressure control maintains the the maximum stability property by Proposition 1.

4.3.1 Travel time function

Besides function (4.45), there are functions that represent the delay for a turning movement, but most of these functions are not monotone increasing with the queue length. However, we do not rule out the possibility of achieving maximum throughput when using these delay functions. To define the delay function, we categorize vehicles in queue $x_{ij}(t)$ based their delays, denoted by $\omega_{ij}(t)$. The delay ω ranges from 1 to ∞. Let $x_{ij}^{\omega}(t)$ be the number of vehicle in queue $x_{ij}(t)$ with a delay of ω. The term $\sum_{\omega=1}^{\infty} \omega x_{ij}^{\omega}(t)$ represents the total delay for vehicles in a queue, which is the summation of the product of different values of delays and the numbers of vehicles with corresponding delays. As shown in equation (4.46), the

average delay of vehicles in queue (i, j) is the total delay divided by the queue length.

$$\omega_{ij}(t) = \frac{\sum_{\omega=1}^{\infty} \omega x_{ij}^{\omega}(t)}{x_{ij}(t)} \tag{4.46}$$

It is obvious that the travel time function is not an increasing function with respect to $x_{ij}(t)$ because the direction of its change over time depends on the proportion of vehicles with different waiting times. For example, at time t, suppose queue (i, j) has two vehicles with waiting times of three time steps, so $x_{ij}^3(t) = 2$. At time t, the average delay for the queue is $\omega_{ij}(t) = \frac{3\times 2}{2} = 3$. At time $t + 1$, suppose that one vehicle leaves the queue and two vehicles enter the queue. Two new vehicles both have a waiting time of one time step while the vehicle already present at the previous time step has a waiting time of four time steps, so $x_{ij}^4(t + 1) = 1, x_{ij}^1(t + 1) = 2$. The average delay at time $t + 1$ is $\omega_{ij}(t + 1) = \frac{2+4}{3} = 2$. Here the queue length increases by one vehicle $(x_{ij}(t) = 2 \rightarrow x_{ij}(t + 1) = 3)$ but the average delay decreases by one time step $(\omega_{ij}(t) = 3 \rightarrow \omega_{ij}(t + 1) = 2)$.

4.4 Numerical experiments

In this section, we study the effects of using different functions in the max-pressure control. Although maximum stability was established analytically, the differences in the constant B could affect the average delay. The experiment uses the Downtown Austin network which has 171 zones, 546 intersections, and 1237 links. The simulation is run under 30 total demand rates ranging from 500 vehicle per hour to 35000 vehicles per hour. We assume that the proportion of demand for each origin-destination pair remained fixed despite increasing total demand rates. The network was calibrated to match Austin traffic data in 2011 by the Network Modeling Center with a calibrated morning peak demand rate of 28111 vehicles per hour. The simulation time for each run is set to be 3 hours. During the simulation, the total queue length in the network for each 15-min time interval, average vehicle travel time,

and average vehicle delay are recorded.

In simulations, we assume that routes are assigned to vehicles before their departure and that route choice is fixed. The average probability of each path to be picked by vehicles is calculated in advance based on dynamic user equilibrium solved with fixed intersection control. We use the path probabilities from dynamic user equilibrium (Chiu et al., 2011) because traffic flows are more evenly distributed among paths, which is more realistic. However, because the store-and-forward queueing model assumes fixed average turning proportions $\bar{\mathbf{p}}$, we do not solve dynamic traffic assignment with max-pressure control. Dynamic traffic assignment is only used to find realistic values for $\bar{\mathbf{p}}$.

The turning proportion of each turning movement is calculated based on the probability of each path. Let $\bar{d}_{rs}$ be the total average demand between origin r and destination s. Let Π be the set of all paths. The probability that path $\pi_{rs} \in \Pi$ picked by a vehicle from r to s is denoted ρ_{rs}^{π}. The path flow h^{π} can be calculated with $h^{\pi} = d_{rs}\rho_{rs}^{\pi}$. Let $\delta_i^{\pi} \in \{0, 1\}$ indicate whether link i is on path π. Then the average turning proportion of a turning movement $\bar{p}_{ij}$ can be calculated using the path flow:

$$\bar{p}_{ij} = \frac{\sum_{\pi \in \Pi} h^{\pi} \delta_i^{\pi} \delta_j^{\pi}}{\sum_{\pi \in \Pi} h^{\pi} \delta_i^{\pi}} \tag{4.47}$$

After calculating the average turning proportions of each turning movement, four functions are used in simulation experiments to calculate the weights of turning movements:

- the queue length: $x_{ij}(t)$
- the travel time function: $\frac{x_{ij}(t)}{Q_{ij}(t)}$
- the average delay function: $\frac{\sum_{\omega=1}^{\infty} \omega x_{ij}^{\omega}(t)}{x_{ij}(t)}$
- the total delay function: $\sum_{\omega=1}^{\infty} \omega x_{ij}^{\omega}(t)$

The queue length and delay information are collected in real time during the simulation based on the information of all waiting vehicles. The optimal intersection control is calculated in every time step using the collected information.

4.4.1 Queue length stability

If the max-pressure control is able to stabilize the queue length, the queue length will fluctuate around a constant value after the simulated network has become sufficiently saturated. Otherwise, the queue length will continuously increase from the beginning to the end of the simulation. Figure 4.1 shows an example of a stable queue and an unstable queue. The total queue lengths $\sum_{(i,j)\in\mathcal{M}} x_{ij}(t)$ are recorded in experiments using the travel time function $x_{ij}(t)/Q_{ij}$ with demand rates of 23000 and 35000 vehicles per hour respectively. Under the demand rate of 23000 vehicles per hour, the total queue length is constant at 800 vehicles after 400 seconds in the simulation. Under the demand rate of 35000 vehicles per hour, the total queue length is always increasing.

Figure 4.2 shows the variation of total queue length in the network. In this figure, scenarios using the queue length function and the travel time function ($x_{ij}(t)/Q_{ij}(t)$) stabilizes the queue length. Scenarios using the total delay and the average delay functions cannot stabilize the queue length, and the average value and the variance of queue lengths in these two scenarios are also larger than the scenarios using the queue length function and the travel time function. After 8000 seconds, differences between the total queue lengths of scenarios using the queue length function and the travel time function and scenarios using two types of delay functions becomes larger.

4.4.2 Average travel time

Figure 6.11 shows vehicle average travel times when using different functions in the max-pressure control. As the demand increases, vehicles spend more time on the link and at

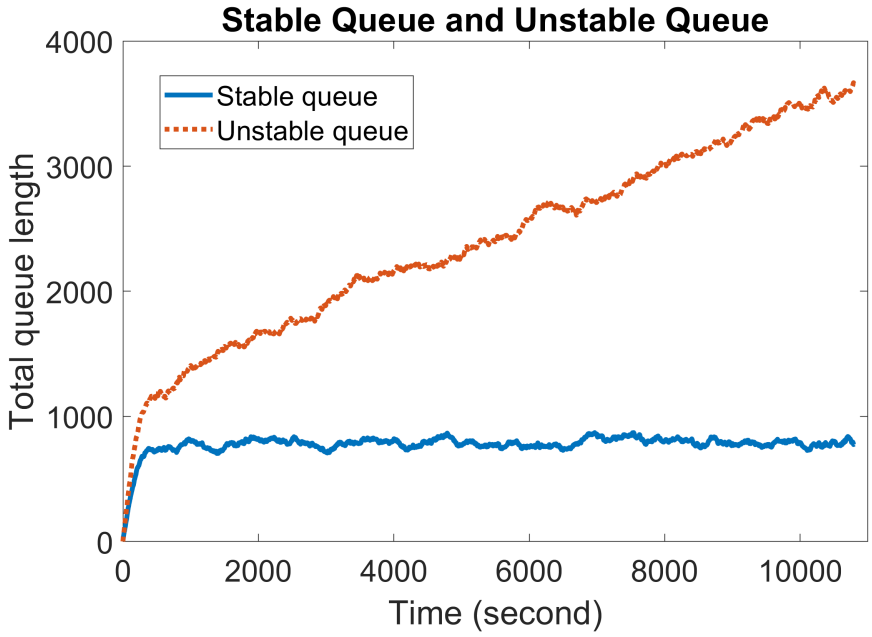


Figure 4.1: An example of stable queue and unstable queue

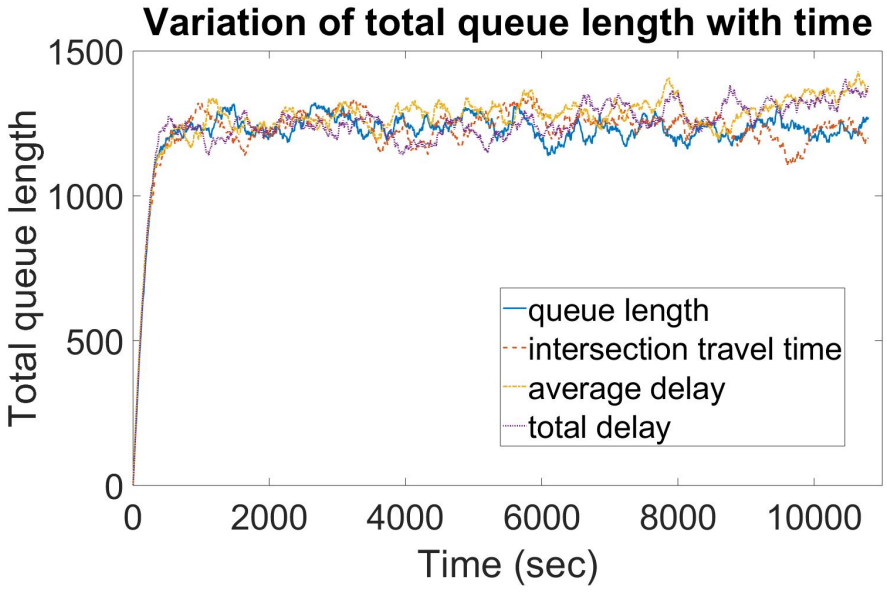


Figure 4.2: Queue lengths of scenarios with four functions tested

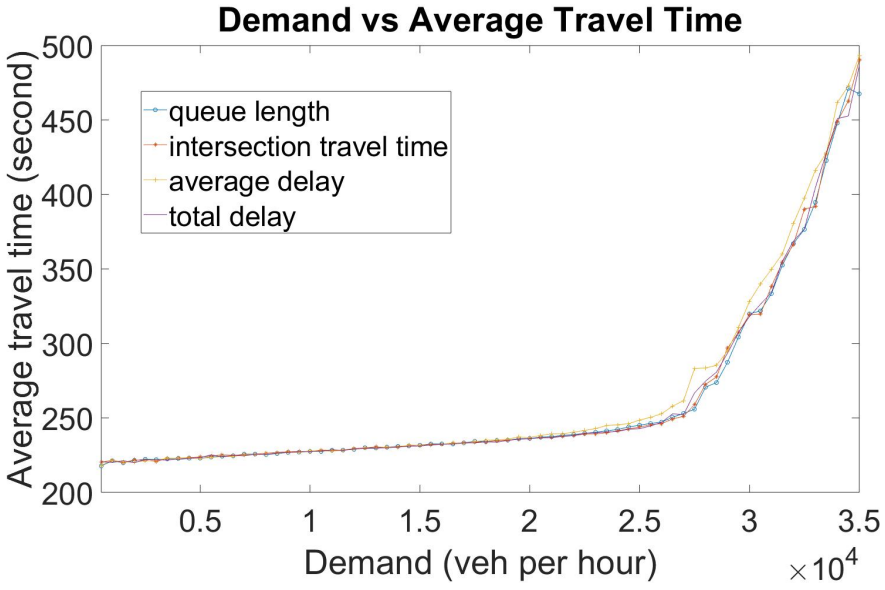


Figure 4.3: Average travel times of scenarios with four functions tested

the intersection, so the average travel time of vehicles increases. Because all these four scenarios use the same turning proportions, the link travel times of vehicles before they stop at the intersection are the same. Therefore, their only difference is the waiting time at the intersection, which can be reflected by the vehicle travel time. The average travel time of the scenario using the average delay function is larger than all the other scenarios when the demand is larger than 23000 vehicles per hour. The max-pressure controls using the queue length function, the travel time function ($x_{ij}(t)/Q_{ij}(t)$), and the total delay function have similar average travel times.

4.4.3 Total queue length

Figure 4.4 shows the total queue length in the network with different demand rates. When the demand increases, the total queue length shows a similar trend as the average travel time. An increase of the demand rate results an increase of the vehicle waiting time at the intersection, so the total queue length at the intersection increases. The scenario using the

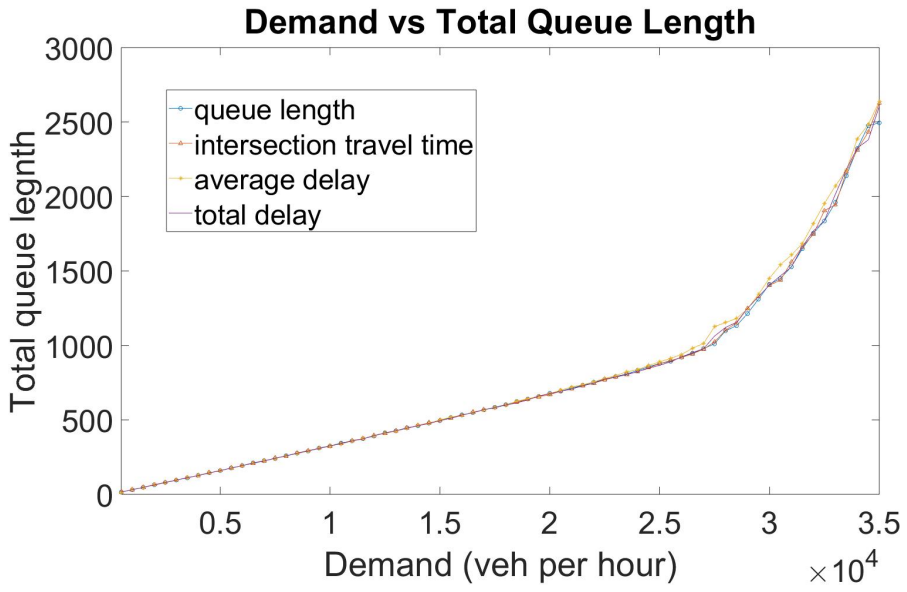


Figure 4.4: Total queue lengths of scenarios with four functions tested

average delay has larger total queue length than other three scenarios when the demand is larger than 27000 vehicle per hour. The scenario using the total delay has slightly larger total queue length than scenarios using the queue length function and the travel time function ($\frac{x_{ij}(t)}{Q_{ij}(t)}$).

Both the average travel times (Figure 6.11) and the total queue length (Figure 4.4) increase with the demand rate because the average service time of the system and the number of vehicles in the system are highly related. However, in Figure 6.11, the corresponding demand when the curve of the average delay increases with a larger speed than other curves is 22000 vehicles per hour while that demand in Figure 4.4 is 25000 vehicles per hour.

4.4.4 Data collection for weight calculation

This section discusses the data collection for road sections that are not equipped with loop detectors or video cameras so that the queue length cannot be directly measured. Most online traffic information is collected based on crowd-sourced GPS data. For example,

Google Maps use the anonymized traffic data collected from drivers who are using Google Maps on smartphones. Using this method, we can obtain measurements for travel speeds or travel times rather than traffic densities or queue length because only part of vehicles is detected. However, these data are aggregated by links so it is hard to get the travel time or the travel speed for each turning movement separately.

Compared with all the other functions, the input data for the average travel time function is most easy to get. If only the online traffic information is available, we can get the travel time of a path that includes the turning movement, and then gets travel times of two routes that end before the intersection or starts after the intersection. The average delay at the intersection can be calculated by using the total path travel time subtracting by travel times of two shorter paths. If the intersection is able to get the trajectory data from vehicles with data collection equipment, traffic state estimation algorithms can be applied to get the average travel time for a turning movement.

4.5 Conclusions

To incorporate the travel time measurement in the max-pressure control, this section identifies sufficient assumptions for a travel time function to stabilize the queue length and proved stability under these assumptions. These assumptions are relatively mild, i.e. the travel time function should be a continuous and monotone increasing function with respect to the queue length. We propose a travel time function including the queue length and the capacity which is a reasonable approximation of the vehicle travel time at the intersection. Two other functions that represent the total delay and the average delay are considered in the analytical discussion and numerical results, but these two functions are not monotone increasing with respect to the queue length.

Simulations are used to test the properties of the max-pressure control with four functions: the queue length function $x_{ij}(t)$, the travel time function $\frac{x_{ij}(t)}{Q_{ij}(t)}$, the total delay func-

tion, and the average delay function. The vehicle travel time, the total queue length in the network are recorded during the simulation when the demand rate increases from 500 to 35000 vehicle per hour. Based on the simulation results, the max-pressure controls using the queue length function and the travel time function have a larger stable demand than the max-pressure controls using the average delay and the total delay functions. When the demand rate is in the feasible region, the max-pressure controls using these four functions show similar performance, which means they have similar total queue lengths and travel times. When deciding which of these four functions to use in the max-pressure control, we prefer the one with smaller total queue length, smaller total travel time, and larger stabilizable demand rates. If there is little difference between the effect of the max-pressure controls using these four functions, using travel times in max-pressure control is an effective alternative when the queue length data is not available.

Although this section assumes constant turning proportions for each turning movement, in reality route choices depend on travel times. Future work should compare travel times and queue lengths in max-pressure control in networks considering the route choice behaviors of road users and developing methods to obtain the user equilibrium in a network with the max-pressure control. Experimental validation of max-pressure control based on travel times (e.g. using Google Maps data with max-pressure signal timing on actual roads) would advance the implementation of max-pressure control.

Chapter 5

Autonomous Intersection Management with Pedestrians based on Max-Pressure Control

Traditional max-pressure control activates a phase with several movements and notify vehicles through traffic signal lights or transmitting messages using V2I technology. When all vehicles are enabled with autonomous driving technologies, signal-free intersection control algorithms can directly control vehicle movements. For example, AIM algorithms coordinate trajectories of vehicles to prevent collisions and improve intersection efficiency. However, AIM algorithms omit pedestrian access. As coordinated vehicle trajectories are close in time and space, pedestrians can hardly cross an intersection controlled by AIM algorithms if there are no underground tunnels or pedestrian overpass. A convenient way to provide pedestrian access is to add crosswalk activation to the existing AIM algorithm, which will reduce the overall efficiency of vehicles. The addition of the max-pressure control maximizes the throughput of the combined pedestrian and vehicle flows. This chapter introduces the study on the integration of the AIM algorithm and the max-pressure control considering pedestrian access. The effects of vehicle and pedestrian demands and pedestrian waiting time limit on intersection efficiency are tested in this chapter.

5.1 Network model

Consider a traffic network consisting of a road network for vehicles and a sidewalk network for pedestrians. These networks interact at intersections where crosswalks and vehicles can conflict. To distinguish the vehicle network and the pedestrian network, superscripts are added to the notations used in the network model proposed in Chapter 3. For example, vehicle network is represented by $\mathcal{G}^{\text{veh}}(\mathcal{N}^{\text{veh}}, \mathcal{L}^{\text{veh}})$, and pedestrian network is represented by $\mathcal{G}^{\text{ped}}(\mathcal{N}^{\text{ped}}, \mathcal{L}^{\text{ped}})$. Similar to Chapter 3, the link set can be classified into three subsets $\mathcal{L}_{\text{entry}}$, $\mathcal{L}_{\text{int}}$, and $\mathcal{L}_{\text{exit}}$ representing the entry, interval, and exiting links respectively. It is assumed that every link has a free-flow travel time of 1 time step, which means the vehicle link and the pedestrian link may have different lengths if vehicles and pedestrians have different travel speeds.

In this section, it is assumed that an intersection is divided into several conflict regions where trajectories of vehicle movements intersect with each other. Conflict regions are incorporated in the traffic flow model for AIM, which are used to constraint the number of vehicles with conflict trajectories passing the intersection. Let $\mathcal{C}$ be the set of all conflict regions at an intersection. $\mathcal{C}_{ij}$ is the set of conflict regions on the trajectory of turning movement (i, j). Let δ_{ij}^c denote the relation between turning movement (i, j) and conflict region c. If turning movement (i, j) intersects with conflict region c, $\delta_{ij}^c = 1$. Otherwise, $\delta_{ij}^c = 0$.

The pedestrian network $\mathcal{G}^{\text{ped}}(\mathcal{N}^{\text{ped}}, \mathcal{L}^{\text{ped}})$ consists of the set of pedestrian nodes $\mathcal{N}^{\text{ped}}$ and the set of pedestrian links $\mathcal{L}^{\text{ped}}$. The pedestrian link set includes sidewalks and crosswalks. In Figure 5.1, the pedestrian links in green are sidewalks, such as links a, b, c, d, e, f, g, and h. If two sidewalks are directly connected, pedestrians do not need to wait when they walk from one to another. For example, the sidewalk b is connected with sidewalk c and pedestrians at sidewalk b can walk to sidewalk c without stopping. The pedestrian links that intersect with vehicle links are crosswalks, such as links i, j, k, and m. Pedestrians

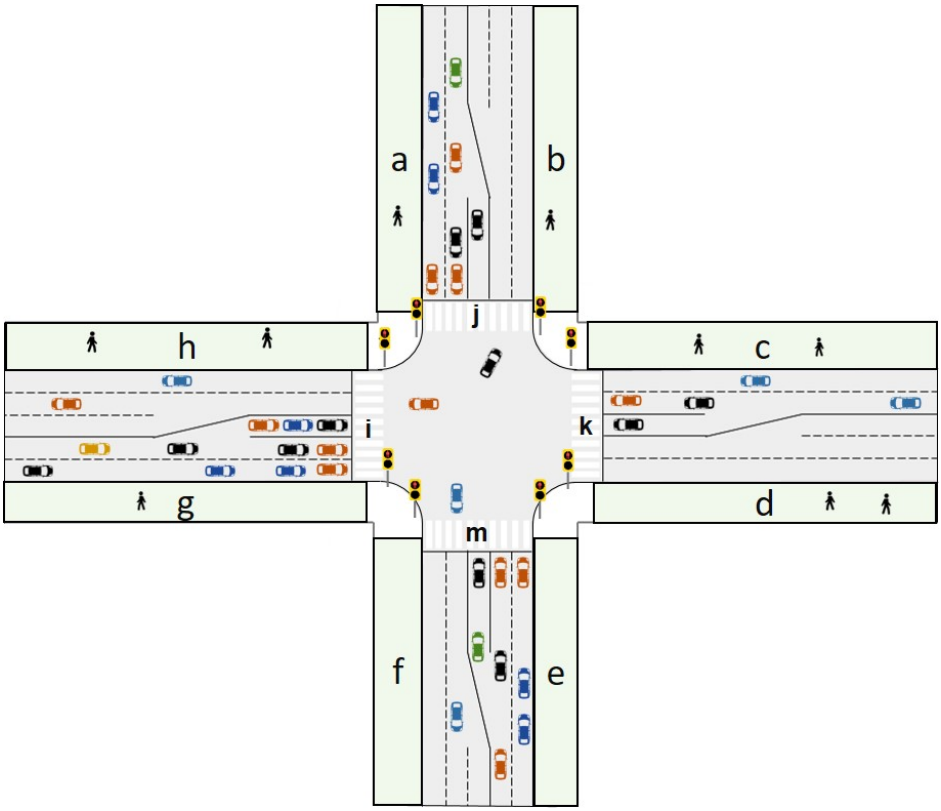


Figure 5.1: Pedestrian network

need to wait for the activation of a crosswalk to cross the street. For example, pedestrians at sidewalk b should wait at crosswalk k if they want to go to sidewalk e until the pedestrian signal turns green. Let $\mathcal{W}$ denote the set of all crosswalks. A pair of adjacent pedestrian links can define a pedestrian movement. A pedestrian movement whose direction is toward a crosswalk is restricted by the actuation of the crosswalk. For example, pedestrian movement (b, k) is restricted by the pedestrian signal on crosswalk k but (b, c) is not.

Considering the interaction between crosswalks and vehicles, δ_{ij}^m is used to indicate if the trajectory of turning movement (i, j) intersects with crosswalk m. If they have an intercept, $\delta_{ij}^m = 1$, otherwise, $\delta_{ij}^m = 0$.

5.1.1 Queue evolution

The queue evolution in this network also follows the same form of the point queue model proposed in Section 3.2. The superscript z is added to variables to indicate the mode type, as shown in equations (5.1), (5.2), (5.3), and (5.4).

If the link is an internal or exit link, then the queue evolution follows equation (5.1).

$$x_{ij}^z(t+1) = x_{ij}^z(t) - y_{ij}^z(t) + \sum_{h \in \Gamma_i^-} y_{hi}^z(t) p_{ij}(t)$$
$$\forall i \in \mathcal{L}_{\text{int}}^z \cup \mathcal{L}_{\text{exit}}^z, j \in \Gamma_i^+, z \in \{\text{veh}, \text{ped}\} \quad (5.1)$$

If the link is an entry link, then the queue evolution follows equation (5.2).

$$x_{ij}^z(t+1) = x_{ij}^z(t) - y_{ij}^z(t) + d_i^z(t) p_{ij}(t) \qquad \forall i \in \mathcal{L}_{\text{entry}}^z, j \in \Gamma_i^+, z \in \{\text{veh}, \text{ped}\} \quad (5.2)$$

The vehicle flow $y_{ij}(t)$ is calculated at every time step based on intersection control.

$$y_{ij}^{\text{veh}}(t) = \min\left\{Q_{ij}(t) s_{ij}^{\text{veh}}(t), x_{ij}(t)\right\} \qquad \forall i, j \in \mathcal{L}^{\text{veh}} \quad (5.3)$$

For pedestrian movement (i, j), if both link i and link j are sidewalks, s_{ij}^{ped} is always 1. If link j is a crosswalk, $s_{ij}^{\text{ped}}(t)$ depends on the intersection control. All pedestrian movements heading for the same crosswalk have the same pedestrian signal control, which is $s_{ij}^{\text{ped}}(t) = s_{hj}^{\text{ped}}(t) = s_j^{\text{ped}}(t), \forall j \in \mathcal{W}, \forall h, i \in \Gamma_j^-, h \neq i$.

$$y_{ij}^{\text{ped}}(t) = \min\left\{Q_{ij}(t)s_{ij}^{\text{ped}}(t), x_{ij}(t)\right\} \qquad \forall i, j \in \mathcal{L}^{\text{ped}} \tag{5.4}$$

5.1.2 Estimating the pedestrian queue length

Unlike automated vehicles, which are assumed to communicate wirelessly with the intersection, pedestrians may only be able to indicate their presence through the crosswalk button. Consequently it is difficult to count the number of waiting pedestrians, which requires an estimation of the waiting queue. The estimated pedestrian queue length is used to calculate the weight for pedestrian turning movements in the max-pressure control and activate crosswalks.

The activation of crosswalk m is represented by $s_m(t)$. When the pedestrian signal is activated at time t, $s_m^{\text{ped}}(t) = 1$. Otherwise, $s_m^{\text{ped}}(t) = 0$. The activation of the pedestrian signal is related to the pedestrian queue, and we want the pedestrian queue length to be bounded. To estimate the pedestrian queue, the pedestrian waiting time should be recorded.

Let $\hat{\tau}_m(t)$ be the waiting time of pedestrian at crosswalk m since the last actuation of the pedestrian signal. $\hat{\tau}_m(t+1)$ can be updated with equation (5.5).

$$\hat{\tau}_m(t+1) = \begin{cases} \hat{\tau}_m(t) + 1, & \text{if } s_m^{\text{ped}}(t) = 0 \wedge \sum_{l \in \Gamma_m^-} x_{lm}(t) > 0 \\ 0, & \text{if } s_m^{\text{ped}}(t) = 1 \vee \sum_{l \in \Gamma_m^-} x_{lm}(t) = 0 \end{cases} \tag{5.5}$$

If the pedestrian signal is not activated at the last time step, and there are waiting pedestrians, then the waiting time will increase by one. If the pedestrian signal is activated at the previous time step or if there is no pedestrian going to cross the road, the waiting time is

set to be 0. Based on this model, the waiting time of a pedestrian queue is determined by the pedestrian with the longest waiting time.

After the estimation of the pedestrian waiting time, the estimated pedestrian queue $\hat{x}_{ij}^{\text{ped}}$ can be calculated using equation (5.6).

$$\hat{x}_{ij}^{\text{ped}}(t+1) = \hat{x}_{ij}^{\text{ped}}(t) - \hat{y}_{ij}^{\text{out}}(t) + \hat{y}_{ij}^{\text{in}}(t) \tag{5.6}$$

Because of the difficulty to directly measure the queue length at crosswalk j, the estimated value instead of the actual value of the queue length is used. In equation (5.6), the estimated pedestrian queue $\hat{x}_{ij}^{\text{ped}}(t+1)$ is the estimated pedestrian queue $\hat{x}_{ij}^{\text{ped}}(t)$ added to the estimated pedestrian entering flow $\hat{y}_{ij}^{\text{in}}(t)$ and minus the estimated pedestrian exiting flow $\hat{y}_{ij}^{\text{out}}(t)$. The estimated entering flow is $\hat{y}_{ij}^{\text{in}}(t) = \hat{\tau}_j(t)\bar{u}_{ij}$. $\hat{\tau}_j$ is the waiting time since the last activation of the pedestrian signal at crosswalk j and $\bar{u}_{ij}$ is the mean arrival rate of pedestrians that are from pedestrian link i to crosswalk j and is assumed to be exogenous. The estimated exiting flow is $\hat{y}_{ij}^{\text{out}}(t) = \min(\hat{x}_{ij}^{\text{ped}}(t), Q_{ij}s_j^{\text{ped}}(t))$, which is the minimum value between the estimated pedestrian queue length $\hat{x}_{ij}^{\text{ped}}(t)$ and the product of the capacity and the crosswalk control $s_j^{\text{ped}}(t)$. If the crosswalk is not activated, the estimated exiting flow is zero. If the crosswalk is activated and the estimated pedestrian queue length at time t does not exceed the capacity, all potential pedestrians can cross the street at the current time step. Otherwise, the number of supposed pedestrians that can pass the street is restricted by the capacity. We assume that the expected value of the difference between the estimated queue length and the actual queue length is bounded by $\omega \geq 0$, i.e. $\mathbb{E}\left[|x_{ij}^{\text{ped}}(t) - \hat{x}_{ij}^{\text{ped}}(t)|\right] \leq \omega$. In the perspective of queuing theory, the average queue length should be the product of the arrival rate and the waiting time. As the estimated queue length is calculated using the measured waiting time and the average arrival rate, the expectation of the estimated queue length and the actual queue length should be equal in the long term.

5.2 Max-pressure control policy

As max-pressure control has the property of maximizing the throughput at the network-level (Varaiya, 2013), this chapter uses a max-pressure algorithm to calculate how many vehicles at the intersection should be served at every time step and the actuation of the crosswalk. The weight function is defined in equation (5.7) for both vehicles and pedestrian movements.

$$w_{ij}^{z}(t) = x_{ij}^{z}(t) - \sum_{k \in \Gamma_j^+} x_{jk}^{z}(t) p_{jk}(t) \qquad z \in \{\text{veh}, \text{ped}\} \tag{5.7}$$

The calculation of the weights for pedestrian queues uses the estimated queue length in equation (5.6). The difference between the weights calculated by the actual and the estimated pedestrian queue length is also assumed to be bounded, which is $|w_{ij}^{\text{ped}}(t) - \hat{w}_{ij}^{\text{ped}}(t)| \leq \beta$.

After calculating the weight for each movement, a mixed-integer linear program is used to calculate the intersection control strategy, as shown in equation (5.8). In this program, the intersection is divided into several conflict regions and each of them has its capacity. Figure 5.2 shows an intersection with four conflict regions (A, B, C, and D). The trajectory of a vehicle will pass one or several conflict regions. We use $\mathcal{C}_{ij}$ to denote the set of conflict regions passed by vehicle movement (i, j). For example, the northbound left-turn movement passes through conflict regions A, C, and D. The capacity of conflict region c is Q_c and is determined by the capacities of turning movements that pass through conflict region, which is $Q_c = \max_{\{(i,j)|c \in \mathcal{C}_{ij}\}}\{Q_{ij}\}$. The total amount of vehicles passing through a conflict region per time step is restricted by the capacity of the conflict region.

$$\max \quad \sum_{(i,j)\in\mathcal{M}} w_{ij}^{\text{veh}}(t) y_{ij}(t) \quad + \quad \sum_{n\in\mathcal{W}, m\in\Gamma_n^-} w_{mn}^{\text{ped}}(t) s_{mn}^{\text{ped}}(t) Q_{mn}^{\text{ped}}(t) \tag{5.8a}$$

$$\text{s.t.} \quad y_{ij}(t) \leq Q_{ij}(t)(1 - s_{mn}^{\text{ped}}(t)\delta_{ij}^{n}) \quad \forall (i,j) \in \mathcal{M}, \forall n \in \mathcal{W}, \forall m \in \Gamma_n^- \tag{5.8b}$$

$$\sum_{(i,j)\in\mathcal{M}} y_{ij}(t)\delta_{ij}^{c} \leq Q_c \qquad \forall c \in \mathcal{C} \tag{5.8c}$$

$$y_{ij}(t) \leq x_{ij}(t) \qquad \forall (i,j) \in \mathcal{M} \tag{5.8d}$$

$$s_{mn}^{\text{ped}}(t) \in \{0,1\} \qquad \forall n \in \mathcal{W}, \forall m \in \Gamma_n^- \tag{5.8e}$$

$$s_{mn}^{\text{ped}}(t) = s_{ln}^{\text{ped}}(t), \qquad \forall m,l \in \Gamma_n^-, m \neq l \tag{5.8f}$$

$$y_{ij}(t) \geq 0 \qquad \forall (i,j) \in \mathcal{M} \tag{5.8g}$$

The max-pressure control aims to optimize the total pressure. y_{ij} represents the number of vehicles in turning movement (i,j) that is allowed to move, which should be constrained by the capacity at the conflict region and the waiting vehicles at the entry approach. s_{mn}^{ped} controls the activation of pedestrian movement (m,n). The value for s_{ij} in equation (5.3) can be calculate using $s_{ij} = y_{ij}/Q_{ij}$. Let s_r^* denote the max-pressure control at intersection r in the network, which is $s_r^* = \arg\max_{y_{ij}, s_{mn}^{\text{ped}}} \left\{ \sum_{(i,j)\in\mathcal{M}} w_{ij}^{\text{veh}} y_{ij} + \sum_{(m,n)\in\mathcal{W}} w_{mn}^{\text{ped}} s_{mn}^{\text{ped}} Q_{mn}^{\text{ped}} \right\}$.

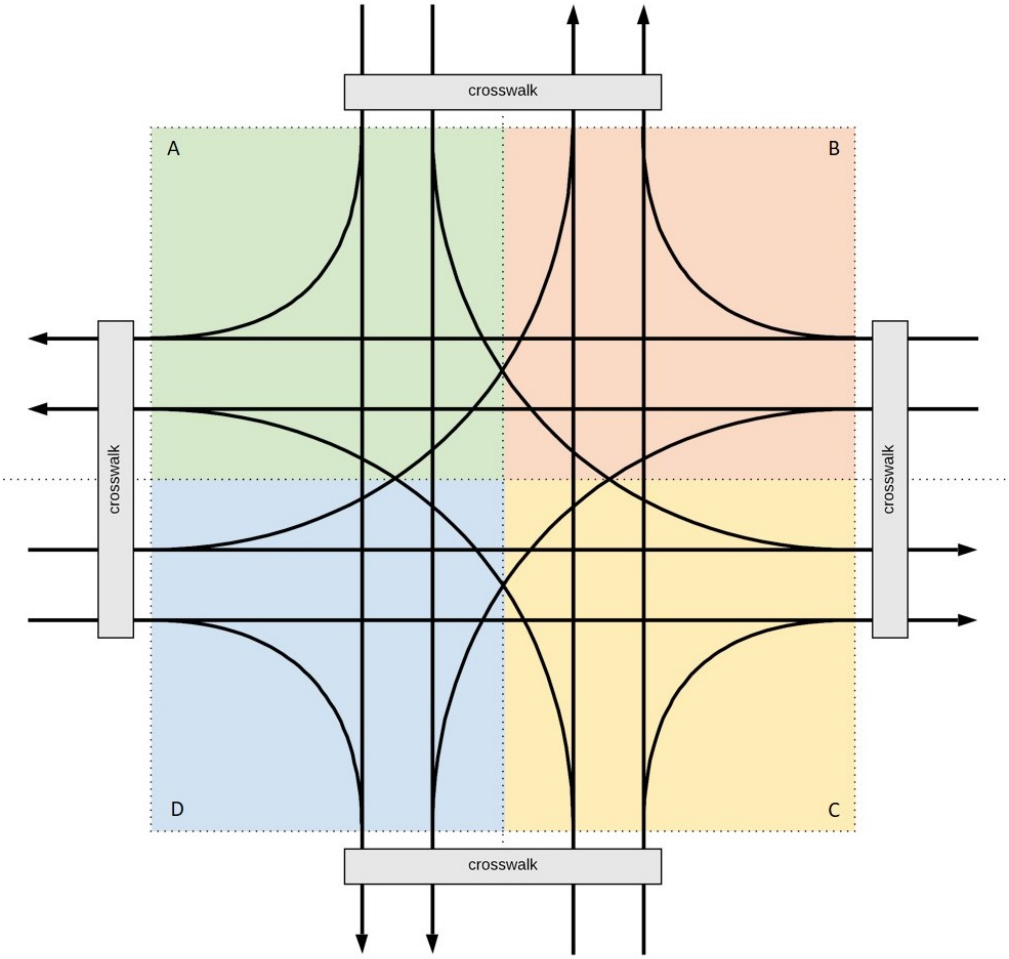


Figure 5.2: Conflict region model of AIM

5.3 Stability region

To derive the stability region of the demand, the average flow rate of both vehicle and pedestrian movements should be calculated using the average demand and average turning proportion that is defined in Section 3.1. The calculation of the average flow rate for each movement is defined in Section 3.3.

The activation of a movement (i, j) at time step t is represented by $s_{ij}(t)$. For vehicles, $s_{ij}(t)$ is a fraction between 0 and 1, which represents the percentage of time used for activating turning movement (i, j). For example, if $s_{ij}(t) = 0.5$ and the time step is 30 seconds, then turning movement (i, j) is allowed to move for 15 seconds in this time step. For pedestrians, it is a binary variable whose value is either 0 or 1 because it is assumed that the pedestrian can use the entire time interval if the pedestrian signal is activated. The turning flow or pedestrian flow $y_{ij}(t)$ is the minimum between the product of the capacity and the movement activation $Q_{ij}s_{ij}(t)$ and the current queue length $x_{ij}(t)$. Equation (5.9) shows the constraint applying to the activation of movements at an intersection. The sum of s_{ij} should be less than or equal to 1, because the sum of percentages of time occupied by any turning movement should be less than or equal to 100%.

$$\sum_{(i,j)\in\mathcal{M}} s_{ij}(t)\delta_{ij}^{c} \leq 1, \forall c \in \mathcal{C} \tag{5.9}$$

Except for variables for movement activation, the notations for matrices of intersection control are the same as what are defined in Section 3.4. For any given intersection control sequence, the long-term average time used for serving turning movement (i, j) and pedestrian movement (i, j) is calculated by equation (3.5). The requirement for intersection control to accommodate demand is given in equation (3.6). The set $\mathcal{D}$ and its interior $\mathcal{D}^o$ are the same as those in Section 3.5.

5.3.1 Stability of the control algorithm

This section considers the queue length stability for the combined flow of vehicles and pedestrians, which is defined in equation (5.10). The stability is proved when max-pressure control is used.

$$\limsup_{T} \frac{1}{T} \sum_{t=0}^{T} \mathbb{E}\left[\sum_{(i,j)\in\mathcal{L}^{\text{veh}}\cup\mathcal{L}^{\text{ped}}} x_{ij}(t)\right] < \infty, \forall T \in \{0, ..., \infty\} \tag{5.10}$$

Lemma 5. *If* $\mathbb{E}\left[|x_{mn}^{\text{ped}}(t) - \hat{x}_{mn}^{ped}|\right] \leq \omega$*, then expectation of the the difference between the optimal solutions of two programs with objective functions* $\hat{O}$ *and* O *for an intersection is also bounded, where* $\hat{O} = \max \sum_{ij} w_{ij}^{\text{veh}} y_{ij} + \sum_{mn}(\hat{x}_{mn}^{\text{ped}} - \sum_o \hat{x}_{no}^{\text{ped}} p_{no}) y_{mn}$ *and* $O = \max \sum_{ij} w_{ij}^{\text{veh}} y_{ij} + \sum_{mn}(x_{mn}^{\text{ped}} - \sum_o x_{no}^{\text{ped}} p_{no}) y_{mn}$.

Proof. The general form of the mixed integer program in equation (5.8) can be expressed as:

$$\begin{array}{ll} \max & h^\intercal \mathbf{y} + c^\intercal \mathbf{S}^{\text{ped}} \\ \text{s.t.} & A\mathbf{y} + G\mathbf{S}^{\text{ped}} \leq b \\ & \mathbf{y} \geq 0 \\ & \mathbf{S}^{\text{ped}} \geq 0 \\ & \mathbf{y} \in \mathbb{R}^{|\mathcal{M}|} \\ & \mathbf{S}^{\text{ped}} \in \{0,1\}^{|\mathcal{W}|} \end{array}$$

In the general form, $\mathbf{y}$ is a vector including decision variables y_{ij} for all vehicle turning movement. $\mathbf{S}^{\text{ped}}$ is a vector including decision variables S_{mn}^{ped} for all crosswalks. Changing the weights of the pedestrian queue is actually changing the cost c in the objective function. The vector c is replaced by $c + \Delta c$ with $\Delta c_{ij} \leq \beta Q_{ij}^{\text{ped}}$. If adding Δc does not change the optimal solution, then $O = \hat{O}$. Even if adding Δc does change the optimal solution, $|O - \hat{O}|$ is also bounded because all constraints build a feasible set which can be enclosed by a polyhedron. As the size of the polyhedron is limited, $|O - \hat{O}|$ can always be bounded by a value, denoted by ξ. □

Proposition 2. *If the demand vector $\bar{\mathbf{d}} \in \mathcal{D}^o$, this max-pressure control is stabilizing.*

If the max-pressure control is stabilizing, the queue length for each turning movement and at each crosswalk will remain bounded in expectation. Equivalently, the control is throughput optimal.

Proof. To calculate the queue length at time $t+1$, we apply the point queue model shown in Equations (5.1) and (5.2).

$$x_{ij}(t+1) = x_{ij}(t) - \min\left(Q_{ij}s_{ij}(t), x_{ij}(t)\right) + \sum_{h\in\Gamma_i^-} \min\left(Q_{hi}s_{hi}(t), x_{hi}(t)\right) p_{ij}(t) \quad \forall i \in \mathcal{L}_{\text{int}}, j \in \Gamma_i^+ \tag{5.12}$$

$$x_{ij}(t+1) = x_{ij}(t) - \min\left(Q_{ij}s_{ij}(t), x_{ij}(t)\right) + d_i(t)p_{ij}(t) \quad \forall i \in \mathcal{L}_{\text{entry}}, j \in \Gamma_i^+ \tag{5.13}$$

Then we get the difference in the queue length between two consecutive time steps:

$$\Delta_{ij} = x_{ij}(t+1) - x_{ij}(t) = -\min\left(Q_{ij}s_{ij}(t), x_{ij}(t)\right) + \sum_{h\in\Gamma_i^-} \min\left(Q_{hi}s_{hi}(t), x_{hi}(t)\right) p_{ij}(t) \quad \forall i \in \mathcal{L}_{\text{int}}, j \in \Gamma_i^+ \tag{5.14}$$

$$\Delta_{ij} = -\min\left(Q_{ij}s_{ij}(t), x_{ij}(t)\right) + d_i(t)p_{ij}(t) \quad \forall i \in \mathcal{L}_{\text{entry}}, j \in \Gamma_i^+ \tag{5.15}$$

Let $X(t)$ be a vector including all queue lengths of all vehicle movements and pedestrian movements. The used Lyapunov function is $|X(t)|^2 = \sum_{(i,j)\in\mathcal{M}} \left(x_{ij}(t)\right)^2$. Then the Lyapunov drift is derived in equation (5.16)

$$|X(t+1)|^2 - |X(t)|^2 = |X(t)+\Delta|^2 - |X(t)|^2 = 2X(t)^\intercal\Delta + |\Delta|^2 \tag{5.16}$$

The first term in the Lyapunov drift is:

$$\begin{aligned}
2X(t)^{\intercal}\Delta = & -2\sum_{i\in\mathcal{L}}\sum_{j\in\Gamma_i^+} x_{ij}(t)\min\left(Q_{ij}s_{ij}(t), x_{ij}(t)\right) \\
& + 2\sum_{h\in\Gamma_i^-}\sum_{i\in\mathcal{L}}\sum_{j\in\Gamma_i^+} x_{ij}(t)\min\left(Q_{hi}s_{hi}(t), x_{hi}(t)\right)p_{ij}(t) \\
& + 2\sum_{i\in\mathcal{L}_{\text{entry}}}\sum_{j\in\Gamma_i^+} x_{ij}(t)\left(-\min\left(Q_{ij}s_{ij}(t), x_{ij}(t)\right) + d_i(t)p_{ij}(t)\right) \\
= & \; 2\sum_{i\in\mathcal{L}_{\text{int}}\cup\mathcal{L}_{\text{entry}}}\sum_{j\in\Gamma_i^+}\min\left(Q_{ij}s_{ij}(t), x_{ij}(t)\right)\left(-x_{ij}(t) + \sum_{k\in\Gamma_j^+} p_{jk}(t)x_{jk}(t)\right) \\
& + 2\sum_{i\in\mathcal{L}_{\text{entry}}}\sum_{j\in\Gamma_i^+} d_i(t)p_{ij}(t)x_{ij}(t) \quad (5.17)
\end{aligned}$$

Then the random variable p is replaced with its mean value $\bar{p}$:

$$\begin{aligned}
& \mathbb{E}\left[X(t)^{\intercal}\Delta | X(t)\right] \\
& = \sum_{i\in\mathcal{L}_{\text{int}}\cup\mathcal{L}_{\text{entry}}}\sum_{j\in\Gamma_i^+}\mathbb{E}\left[\min\left(Q_{ij}s_{ij}(t), x_{ij}(t)\right)\left(-x_{ij}(t)\right) \mid X(t)\right] \\
& \qquad + \sum_{i\in\mathcal{L}_{\text{int}}\cup\mathcal{L}_{\text{entry}}}\sum_{j\in\Gamma_i^+}\mathbb{E}\left[\min\left(Q_{ij}s_{ij}(t), x_{ij}(t)\right)|X(t)\right]\left(\sum_{k\in\Gamma_j^+}\bar{p}_{jk}x_{jk}(t)\right) \\
& \qquad + \sum_{i\in\mathcal{L}_{\text{entry}}}\sum_{j\in\Gamma_i^+}\mathbb{E}\left[d_i(t)\bar{p}_{ij}x_{ij}(t) \mid X(t)\right] \quad (5.18)
\end{aligned}$$

The turning proportion and the downstream queue length are taken out of the expectation term:

$$\begin{aligned}
& \mathbb{E}\left[\min\left(Q_{ij}s_{ij}(t), x_{ij}(t)\right)\bar{p}_{jk}(t)x_{jk}(t)|X(t)\right] = \\
& \qquad \mathbb{E}[\min\left(Q_{ij}s_{ij}(t), x_{ij}(t)\right)|X(t)]\bar{p}_{jk}x_{jk}(t) \quad (5.19)
\end{aligned}$$

By the definition of the pressure, $-x_{ij}(t) + \sum_{k\in\Gamma_j^+} \bar{p}_{jk}x_{jk}(t) = w_{ij}(t)$, so we obtain:

$$\begin{aligned}
&\mathbb{E}[X(t)^\intercal \Delta | X(t)] \\
&\quad = -\sum_{i\in\mathcal{L}_{\text{int}}\cup\mathcal{L}_{\text{entry}}} \mathbb{E}[\min\left(Q_{ij}s_{ij}(t), x_{ij}\right)|X(t)]w_{ij}(t) + \sum_{i\in\mathcal{L}_{\text{entry}}} \bar{d}_i\bar{p}_{ij}x_{ij}(t) \quad (5.20)
\end{aligned}$$

The last term in equation (5.18) is reformulated as:

$$\begin{aligned}
\sum_{i\in\mathcal{L}_{\text{entry}}} \bar{d}_i\bar{p}_{ij}x_{ij}(t) &= \sum_{i\in\mathcal{L}_{\text{entry}}} f_{ij}x_{ij}(t) \\
&= \sum_{i\in\mathcal{L}_{\text{int}}\cup\mathcal{L}_{\text{entry}}} f_i\bar{p}_{ij}x_{ij}(t) - \sum_{j\in\mathcal{L}_{\text{int}}} f_j\bar{p}_{jk}x_{jk}(t) \\
&= \sum_{i\in\mathcal{L}_{\text{int}}\cup\mathcal{L}_{\text{entry}}} f_i\bar{p}_{ij}x_{ij}(t) - \sum_{j\in\Gamma_i^+}\left[\sum_{i\in\mathcal{L}_{\text{int}}\cup\mathcal{L}_{\text{entry}}} f_i\bar{p}_{ij}\right]\sum_k \bar{p}_{jk}x_{jk}(t) \\
&= \sum_{i\in\mathcal{L}_{\text{int}}\cup\mathcal{L}_{\text{entry}}} f_i\bar{p}_{ij}\left[x_{ij}(t) - \sum_{k\in\Gamma_i^+}\bar{p}_{jk}x_{jk}(t)\right] \\
&= \sum_{i\in\mathcal{L}_{\text{int}}\cup\mathcal{L}_{\text{entry}}} f_i\bar{p}_{ij}w_{ij} \quad (5.21)
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mathbb{E}\left[X(t)^\intercal \Delta | X(t)\right] \\
&= \sum_{i\in\mathcal{L}_{\text{int}}\cup\mathcal{L}_{\text{entry}}} \left(f_i\bar{p}_{ij} - \mathbb{E}\left[\min(Q_{ij}s_{ij}, x_{ij}(t))|X(t)\right]\right)w_{ij}(t) \\
&\quad = \sum_{i\in\mathcal{L}_{\text{int}}\cup\mathcal{L}_{\text{entry}}} \left(f_i\bar{p}_{ij} - \bar{Q}_{ij}s_{ij}(t)\right)w_{ij}(t) \\
&\qquad + \sum_{i\in\mathcal{L}_{\text{int}}\cup\mathcal{L}_{\text{entry}}} \left(\bar{Q}_{ij}s_{ij}(t) - \mathbb{E}\left[\min(Q_{ij}s_{ij}(t), x_{ij}(t))|X(t)\right]\right)w_{ij}(t)
\end{aligned}$$

$\bar{Q}$ is the average value of the random variable Q. $\hat{Q}$ is the maximum value of Q. For vehicle

flow (i,j), $Q_{ij}^{\text{veh}} s_{ij}^{\text{veh}}(t) = y_{ij}^{\text{veh}}(t) \leq x_{ij}^{\text{veh}}(t)$, so $\min\left(Q_{ij}^{\text{veh}} s_{ij}^{\text{veh}}(t), x_{ij}^{\text{veh}}(t)\right) = Q_{ij}^{\text{veh}} s_{ij}^{\text{veh}}(t)$. Then, $\left(\bar{Q}_{ij}^{\text{veh}} s_{ij}^{\text{veh}}(t) - Q_{ij}^{\text{veh}} s_{ij}^{\text{veh}}(t)\right) w_{ij}(t) \leq \bar{Q}_{ij}^{\text{veh}} w_{ij}(t) \leq \bar{Q}_{ij}^{\text{veh}} \hat{Q}_{ij}$. For pedestrian flow (i,j), $s_{ij}^{\text{ped}}(t) \in \{0,1\}$. To get the upper bound, $s_{ij}^{\text{ped}}(t)$ is pulled out.

$$\begin{aligned}\left(\bar{Q}_{ij}^{\text{ped}} s_{ij}^{\text{ped}}(t) - \mathbb{E}\left[\min(Q_{ij}^{\text{ped}} s_{ij}^{\text{ped}}(t), x_{ij}^{\text{ped}}(t))|X(t)\right]\right) w_{ij}(t) \\ = \left(\bar{Q}_{ij}^{\text{ped}} - \mathbb{E}\left[\min(Q_{ij}^{\text{ped}}, x_{ij}^{\text{ped}}(t))|X(t)\right]\right) s_{ij}^{\text{ped}}(t) w_{ij}^{\text{ped}}(t)\end{aligned}$$

When $x_{ij}(t) \geq \hat{Q}_{ij}^{\text{ped}}$, $\left(\bar{Q}_{ij}^{\text{ped}} - \mathbb{E}\left[\min(Q_{ij}^{\text{ped}}, x_{ij}^{\text{ped}}(t))|X(t)\right]\right) = 0$. Otherwise, $\left(\bar{Q}_{ij}^{\text{ped}} - \mathbb{E}\left[\min(Q_{ij}^{\text{ped}}, x_{ij}^{\text{ped}}(t))|X(t)\right]\right) = x_{ij}(t)$, and $x_{ij}(t) w_{ij}^{\text{ped}}(t) \leq \bar{Q}_{ij}^{\text{ped}} \hat{Q}_{ij}^{\text{ped}}$.
Therefore,

$$\sum_{i \in \mathcal{L}_{\text{int}} \cup \mathcal{L}_{\text{entry}}} \left(\bar{Q}_{ij} s_{ij}(t) - \mathbb{E}\left[\min(Q_{ij} s_{ij}(t), x_{ij}(t))|X(t)\right]\right) w_{ij}(t) \leq \sum_{i \in \mathcal{L}_{\text{int}} \cup \mathcal{L}_{\text{entry}}} \bar{Q}_{ij} \hat{Q}_{ij}$$

.

At each time step, an intersection control matrix $s_r(t)$ is selected from the set of signal control matrices. The max-pressure algorithm can get an intersection control matrix with the maximum pressure

$$s^* = \arg\max_{s_{ij}, s_n} \left\{ \sum_{(i,j) \in \mathcal{M}} s_{ij} Q_{ij} w_{ij}^{\text{veh}} + \sum_{n \in \mathcal{W}} s_n Q_n w_n^{\text{ped}} \right\} \tag{5.22}$$

As demand vector $\mathbf{d}$ is in the stability region, the relation between the arrival rate and the intersection control has the relation $f_i \bar{p}_{ij} + \epsilon = \bar{s}_{ij} \bar{Q}_{ij}$ for both vehicle turning movements and pedestrian flows, where $\epsilon > 0$. Inequality (5.23) holds because of Lemma 5.

$$\mathbb{E}\left[\sum_{i \in \mathcal{L}_{\text{int}}^{\text{veh}} \cup \mathcal{L}_{\text{entry}}^{\text{veh}}} [f_i \bar{p}_{ij} - Q_{ij} s_{ij}^*(t)] w_{ij}^{\text{veh}}(t)\right] + \mathbb{E}\left[\sum_{m \in \mathcal{L}_{\text{int}}^{\text{ped}} \cup \mathcal{L}_{\text{entry}}^{\text{ped}}} [f_m \bar{p}_{mn} - Q_{mn} s_n^*(t)] w_{mn}^{\text{ped}}(t)\right]$$

$$\leq \mathbb{E}\left[\sum_{i\in\mathcal{L}_{\text{int}}^{\text{veh}}\cup\mathcal{L}_{\text{entry}}^{\text{veh}}}[f_i\bar{p}_{ij}-Q_{ij}\bar{s}_{ij}^{\text{veh}}]w_{ij}^{\text{veh}}(t)\right]+\mathbb{E}\left[\sum_{m\in\mathcal{L}_{\text{int}}^{\text{ped}}\cup\mathcal{L}_{\text{entry}}^{\text{ped}}}[f_m\bar{p}_{mn}-Q_{mn}\bar{s}_{mn}^{\text{ped}}]w_{mn}^{\text{ped}}(t)\right]$$

$$\leq \mathbb{E}\left[\sum_{i\in\mathcal{L}_{\text{int}}^{\text{veh}}\cup\mathcal{L}_{\text{entry}}^{\text{veh}}}[f_i\bar{p}_{ij}-Q_{ij}\bar{s}_{ij}^{\text{veh}}]w_{ij}^{\text{veh}}(t)\right]+\mathbb{E}\left[\sum_{m\in\mathcal{L}_{\text{int}}^{\text{ped}}\cup\mathcal{L}_{\text{entry}}^{\text{ped}}}[f_m\bar{p}_{mn}-Q_{mn}\bar{s}_{mn}^{\text{ped}}]\hat{w}_{mn}^{\text{ped}}(t)\right]$$
$$+|\mathcal{N}^{\text{ped}}|\xi \quad (5.23)$$

$\bar{s}_{ij}^{\text{veh}}$ and $\bar{s}_{mn}^{\text{ped}}$ correspond to the average actuation rates of the turning movement (i,j) and the crosswalk n of stationary control. If $w_{ij}^{\text{veh}}(t)>0$, then turning movement (i,j) will be activated. So the average actuation rate for the period when turning movement (i,j) is activated times the turning capacity has the relation $Q_{ij}\bar{s}_{ij}^{\text{veh}} > f_i\bar{p}_{ij}+\epsilon$ because $\bar{s}_{ij}^{\text{veh}}$ should satisfy that $Q_{ij}\bar{s}_{ij}^{\text{veh}} > f_i\bar{p}_{ij}$. Otherwise, when $w_{ij}^{\text{veh}}\leq 0$, $Q_{ij}\bar{s}_{ij}^{\text{veh}}=0$ because turning movement (i,j) is not actuated based on max-pressure algorithm. Similarly, for crosswalk n, when $w_{mn}^{\text{ped}}>0$, the average actuation rate times the capacity have the relation $Q_{mn}\bar{s}_{ij}^{\text{ped}}=f_i\bar{p}_{ij}+\epsilon$. Otherwise, $w_{ij}^{\text{ped}}\leq 0$, $Q_{ij}\bar{s}_{ij}^{\text{ped}}=0$.

$$\begin{aligned}
&\sum_{i\in\mathcal{L}_{\text{int}}\cup\mathcal{L}_{\text{entry}}}[f_i\bar{p}_{ij}-Q_{ij}\bar{s}_{ij}(t)]w_{ij}(t)\\
&\qquad\leq -\epsilon\sum_{ij}\max\{w_{ij}^{\text{veh}},0\}+\sum_{ij}f_i\bar{p}_{ij}\max\{-w_{ij}^{\text{veh}},0\}\\
&\qquad\qquad -\epsilon\sum_{mn}\max\{w_{mn}^{\text{ped}},0\}+\sum_{mn}f_m\bar{p}_{mn}\max\{-\hat{w}_{mn}^{\text{ped}},0\}\\
&\qquad\qquad\leq -\epsilon|w_{ij}^{\text{veh}}|-\epsilon|w_{mn}^{\text{ped}}|
\end{aligned}$$

As we assume that $|x_{mn}^{\text{ped}}(t)-\hat{x}_{mn}^{\text{ped}}(t)|\leq\omega$, and $|w_{mn}^{\text{ped}}-\hat{w}_{mn}^{\text{ped}}|\leq\beta$ and based on Lemma 5, we have

$$\sum_{i\in\mathcal{L}_{\text{int}}\cup\mathcal{L}_{\text{entry}}}[f_i\bar{p}_{ij}-Q_{ij}s_{ij}(t)]w_{ij}(t)\leq -\epsilon|w_{ij}^{\text{veh}}|-\epsilon|\hat{w}_{ij}^{\text{ped}}|+|\mathcal{N}^{\text{ped}}|\xi \quad (5.24)$$

The formula used to get the pressure w is a linear function of the matrix X that represent the queue length, so we can find $\eta_1, \eta_2 > 0$ such that $\sum_{ij\in\mathcal{M}} w_{ij}^{\text{veh}} \geq \eta_1 |X^{\text{veh}}(t)|$ and $\sum_{mn\in\mathcal{W}} w_{mn}^{\text{ped}} \geq \eta_2 |X^{\text{ped}}(t)|$. Therefore, we have

$$\begin{aligned}
&-\epsilon|w_{ij}^{\text{veh}}| - \epsilon|\hat{w}_{ij}^{\text{ped}}| + |\mathcal{N}^{\text{ped}}|\xi \\
&\leq -\epsilon\eta_1|X^{\text{veh}}(t)| - \epsilon\eta_2|X^{\text{ped}}(t)| + |\mathcal{N}^{\text{ped}}|\xi \\
&\qquad\qquad \leq \sum_{i\in\mathcal{L}_{\text{int}}\cup\mathcal{L}_{\text{entry}}} \bar{Q}_{ij}\hat{Q}_{ij} - \epsilon\eta_1|X^{\text{veh}}(t)| - \epsilon\eta_2|X^{\text{ped}}(t)| + |\mathcal{N}^{\text{ped}}|\xi \quad (5.25)
\end{aligned}$$

Let $\eta = \min(\eta_1, \eta_2)$ and let $\psi = |\mathcal{N}^{\text{ped}}|\xi$, we have:

$$\mathbb{E}\{X(t)^\intercal \Delta | X(t)\} \leq \sum_{i\in\mathcal{L}_{\text{int}}\cup\mathcal{L}_{\text{entry}}} \bar{Q}_{ij}\hat{Q}_{ij} - \epsilon\eta|X(t)| + \psi \quad (5.26)$$

For $|\Delta|^2$, we have:

$$\begin{aligned}
|\Delta_{ij}| &= |-\min\{Q_{ij}s_{ij}(t), x_{ij}(t)\} + \sum_{h\in\Gamma_i^-} \min\{Q_{hi}s_{hi}(t), x_{hi}(t)\}p_{ij}(t)| \\
&\qquad\qquad \leq \max\{\hat{Q}_{ij}, \sum_{h\in\Gamma_i^-} \hat{Q}_{hi}\} \quad \forall i \in \mathcal{L}_{\text{int}}, j \in \Gamma_i^+ \quad (5.27) \\
|\Delta_{ij}| &= |-\min\{Q_{ij}s_{ij}(t), x_{ij}(t)\} + f_{ij}(t)| \\
&\qquad\qquad \leq \max\{\hat{Q}_{ij}, \hat{d}_{ij}\} \quad \forall i \in \mathcal{L}_{\text{entry}}, j \in \Gamma_i^+ \quad (5.28)
\end{aligned}$$

where $\hat{d}_{ij}$ represent the maximum value of the demand.

Let

$$\gamma_1 = \max\left\{\hat{Q}_{ij}^{\text{veh}}, \sum_{h\in\Gamma_i^-} \hat{Q}_{hi}^{\text{veh}}, \hat{d}_{ij}^{\text{veh}}\right\}, \gamma_2 = \max\left\{\hat{Q}_{ij}^{\text{ped}}, \sum_{h\in\Gamma_i^-} \hat{Q}_{hi}^{\text{ped}}, \hat{d}_{ij}^{\text{ped}}\right\}$$

Therefore, $\sum_{ij} |\Delta_{ij}|^2 \leq \mathcal{N}_1\gamma_1^2 + \mathcal{N}_2\gamma_2^2$, where $\mathcal{N}_1$ is the total number of vehicle movements

and $\mathcal{N}_2$ is the total number of pedestrian movements.

$$\begin{aligned}|X(t+1)|^2 - |X(t)|^2 &= 2X(t)^\intercal \Delta + |\Delta|^2 \\ &\leq 2\left(\sum_{i\in\mathcal{L}_{\text{int}}\cup\mathcal{L}_{\text{entry}}} \bar{Q}_{ij}\hat{Q}_{ij} - \epsilon\eta|X(t)| + \psi\right) + \mathcal{N}_1\gamma_1^2 + \mathcal{N}_2\gamma_2^2 = \kappa - \lambda|X(t)| \end{aligned} \tag{5.29}$$

where $\kappa = 2\sum_{i\in\mathcal{L}_{\text{int}}\cup\mathcal{L}_{\text{entry}}} \bar{Q}_{ij}\hat{Q}_{ij} + \mathcal{N}_1\gamma_1^2 + \mathcal{N}_2\gamma_2^2 + \psi, \epsilon\eta = \lambda$. Now we have:

$$\mathbb{E}\big[|X(t+1)|^2 - |X(t)|^2|X(t)\big] \leq \kappa - \lambda|X(t)| \tag{5.30}$$

If we sum from $t = 1$ to T,

$$\sum_{t=1}^{t=T}\left(\mathbb{E}\big[|X(t+1)|^2 - |X(t)|^2|X(t)\big]\right) \leq \sum_{t=0}^{t=T}(\kappa - \lambda|X(t)|) \tag{5.31}$$

$$\mathbb{E}\big[|X(T+1)|^2 - |X(1)|^2|X(t)\big] \leq T\kappa - \lambda\mathbb{E}\left[\sum_{t=0}^{t=T}|X(t)|\right] \tag{5.32}$$

$$\lambda\mathbb{E}\left[\sum_{t=0}^{t=T}|X(t)|\right] \leq T\kappa + \mathbb{E}\left[|X(1)|^2\right] - \mathbb{E}\left[|X(T+1)|^2\right] \leq T\kappa + \mathbb{E}\left[|X(1)|^2\right] \tag{5.33}$$

$$\frac{1}{T}\mathbb{E}\left[\sum_{t=0}^{t=T}|X(t)|\right] \leq \frac{1}{\lambda}\left(\kappa + \mathbb{E}\left[|X(1)|^2\right]\right) \tag{5.34}$$

□

5.4 Intersection control: AIM-*ped*

In Section 5.2, the max-pressure algorithm is proposed based on the conflict region model of AIM (Levin and Boyles, 2015) and the proposed stability properties are also based on this model. When applied to microscopic simulation, the conflict region model of AIM has

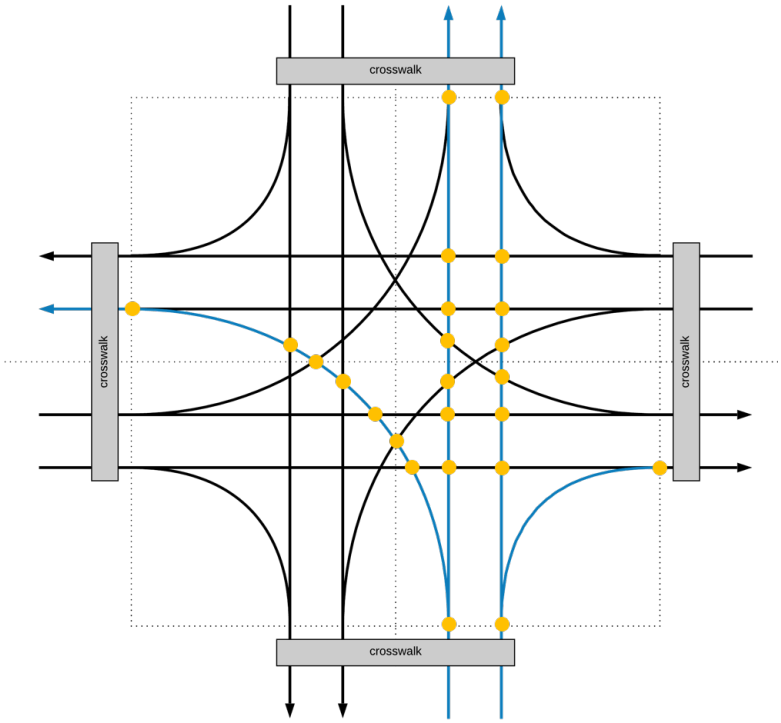


Figure 5.3: Conflict point model of AIM

some limitations. This model only considers the capacity constraint at each conflict region but does not consider the order of arriving vehicles at the intersection. Therefore, the vehicle behavior in this model may violate the first-in-first-out assumption in some conflict regions. For example, if there are three through vehicles from northbound, westbound, and southbound approaches respectively. Assume the intersection has four conflict regions, so each pair of these three vehicles share a conflict region at the intersection. Using this model, it is hard to incorporate time-dependent conflict avoidance within each time step. To add pedestrians to the microscopic simulation with AIM, we create a new algorithm called AIM-*ped* by combining the max-pressure control algorithm proposed in Section 5.2 with the trajectory optimization of Levin and Rey (2017) .

In the model of Levin and Rey (2017), two intersecting vehicle trajectories form a

conflict point. Figure 5.3 shows the conflict points on the trajectories of northbound left-turn, through, and right-turn movements at a standard intersection with four two-lane approaches. The original objective function used in their study is to minimize the exit time of the last vehicle at the intersection. The constraints are used to ensure vehicle trajectories are collision-free. In AIM-*ped*, we control vehicles individually rather than controlling the flow of turning movements and constraints relating vehicles to crosswalks are added to the existing constraints. The objective function is modified to maximize the total pressure:

$$\max \quad \sum_{v \in \mathcal{V}} w_v^{\text{veh}} z_v^{\text{veh}} + \sum_{n \in \mathcal{W}} w_n^{\text{ped}} Q_n^{\text{ped}} z_n^{\text{ped}} \tag{5.35}$$

In equation (5.35), the decision variables are z_v^{veh} and z_n^{ped}, which control the movement of vehicle v, and the actuation of crosswalk n. $\mathcal{V}$ and $\mathcal{W}$ represent the sets of vehicles and crosswalks respectively.

We first show the constraints borrowed from the study of Levin and Rey (2017). Constraint (5.36) requires the calculated departure time of vehicle v to be larger than its earliest possible arrival time e_v at the intersection. γ_v^- is the first conflict point in the path of vehicle v, then $t_v(\gamma_v^-)$ represents the moment when vehicle v enters the intersection.

$$t_v(\gamma_v^-) \geq e_v \quad \forall v \in \mathcal{V} \tag{5.36}$$

Constraint (5.37) guarantees that if two vehicles v and v' share the same entry lane ($\gamma_v^- = \gamma_{v'}^-$), then the vehicle that reach the intersection earlier should also enters the intersection earlier. e_v is the earliest time of vehicle v to reach the intersection and $t_v(\gamma_v^-)$ is the time when vehicle v enters the intersection. $\tau_v(\gamma_v^-)$ is the time that vehicle v occupies the entry point of the intersection on its path.

$$t_v(\gamma_v^-) + \tau_v(\gamma_v^-) \leq t_{v'}(\gamma_{v'}^-) \quad \forall v, v' \in \mathcal{V} : \gamma_v^- = \gamma_{v'}^-, e_v < e_{v'} \tag{5.37}$$

Constraint (5.38) calculates the time that vehicle v spends at conflict point c, denoted by $\tau_v(c)$. $L_v(c)$ is the distance that vehicle v travels around c and $d_v(\gamma_v^-, \gamma_v^+)$ is the travel distance of vehicle v along its path. w is the backward shock wave speed of the related fundamental diagram of this road, and ρ is the set of conflict points on the trajectory of vehicle v. The derivation of this constraint is included in Levin and Rey (2017).

$$\tau_v(c) = \frac{L_v(c)}{w} + \frac{L_v(c)(t_v(\gamma_v^+) - t_v(\gamma_v^-))}{d_v(\gamma_v^-, \gamma_v^+)} \quad \forall v \in \mathcal{V}, \forall c \in \rho_v \tag{5.38}$$

Constraints (5.39) and (5.40) both control the vehicle travel times. Constraint (5.39) sets the upper bound and lower bound of the vehicle travel time through the intersection. $t_v(\gamma_v^+) - t_v(\gamma_v^-)$ is the time that vehicle v uses to pass through the intersection. $\overline{U}_v$ and $\underline{U}_v$ are the maximum and the minimum speeds respectively. Constraint (5.40) calculates the travel time through each conflict point. $t_v(c)$ is the time that vehicle v arrives the conflict point c. $d_v(\gamma_v^-, c)$ is the travel distance from the entry point γ_v^- to conflict point c. This constraint requires that the average travel speed between the entry point and any conflict point on the path should equal the average travel speed between the entry point and the exit point, which pushes the vehicle to keep a constant speed.

$$\frac{d_v(\gamma_v^-, \gamma_v^+)}{\overline{U}_v} \leq t_v(\gamma_v^+) - t_v(\gamma_v^-) \leq \frac{d_v(\gamma_v^-, \gamma_v^+)}{\underline{U}_v} \quad \forall v \in \mathcal{V} \tag{5.39}$$

$$\frac{t_v(c) - t_v(\gamma_v^-)}{d_v(\gamma_v^-, c)} = \frac{t_v(\gamma_v^+) - t_v(\gamma_v^-)}{d_v(\gamma_v^-, \gamma_v^+)} \quad \forall v \in \mathcal{V}, \forall c \in \rho_v \tag{5.40}$$

In constraint (5.41), if vehicles v and v' use the same entry lane ($\gamma_v^- = \gamma_{v'}^-$) and vehicle v arrives earlier, then vehicle v' can only enter each shared conflict point c after vehicle v exits. In constraint (5.42), for two vehicles with conflicting trajectories, variables $\delta_{vv'}$ and $\delta_{v'v}$ are used to represent the order of vehicles entering conflict point c. If vehicle v enters first, $\delta_{vv'}(c) = 1$ and $\delta_{v'v}(c) = 0$. Otherwise, $\delta_{vv'}(c) = 0$ and $\delta_{v'v}(c) = 1$. Constraint (5.43) sets the range of arrival times of two vehicles at each conflict point with conflicting

trajectories. $M_{vv'}$ is a large number. One vehicle can only enter the conflict point c when the other vehicle have passed and variable $\delta_{vv'}$ sets the passing order. If $\delta_{vv'}(c) = 1$, then vehicle v enters the conflict c earlier than vehicle v', then the right hand side equals to 0, and the constraint becomes $t_v(c) + \tau_v(c) \leq t_{v'}(c)$. If $\delta_{vv'}(c) = 0$, the constraint becomes $t_v(c)+\tau_v(c) \leq M_{vv'}$, which means there is no restriction on the arrival times of two vehicle at the conflict point.

$$t_v(c) + \tau_v(c) \leq t_{v'}(c) \quad \forall v, v' \in \mathcal{V} : \gamma_v^- = \gamma_{v'}^-, e_v < e_{v'}, \forall c \in \rho_v \cap \rho_{v'} \tag{5.41}$$

$$\delta_{vv'}(c) + \delta_{v'v}(c) = 1 \quad \forall v, v' \in \mathcal{V} : \gamma_v^- \neq \gamma_{v'}^-, v < v', \forall c \in \rho_v \cap \rho_{v'} \tag{5.42}$$

$$t_v(c) + \tau_v(c) - t_{v'}(c) \leq (1 - \delta_{vv'}(c))M_{vv'} \quad \forall v, v' \in \mathcal{V} : \gamma_v^- \neq \gamma_{v'}^-, \forall c \in \rho_v \cap \rho_{v'} \tag{5.43}$$

$$\delta_{vv'}(c) \in \{0, 1\} \quad \forall v, v' \in \mathcal{V} : \gamma_v^- \neq \gamma_{v'}^-, \forall c \in \rho_v \cap \rho_{v'} \tag{5.44}$$

To integrate the trajectory planning model with max-pressure control, additional constraints are introduced to control the activation of vehicle movements and crosswalks. Constraint (5.45) controls the activation of crosswalks and vehicle movements. z_n^{ped} and z_v^{veh} are binary variables indicating the activation of crosswalk n and vehicle movement v. δ_v^n indicates whether the trajectory of vehicle v intersects with crosswalk n which is determined in advance. If crosswalk n is activated and the trajectory of vehicle v is conflicting with crosswalk n, then this vehicle is not allowed to move at the current time step. Constraint (5.46) relates the activation of two vehicles on the same entry lane. If the preceding vehicle is not allowed to move, the following vehicle is not allowed to move either. Constraint (5.47) plays an important role in relating the max-pressure control with the trajectory planning model because it builds the relationship between decision variables. t is the current time and Δt is the length of the time interval. When vehicle v is not allowed to move at the current time interval and $z_v^{\text{veh}} = 0$, then constraint becomes $t_v(\gamma_v^+) + \tau_v(\gamma_v^+) \leq t + \Delta t + M_v$, which means the exit time of vehicle v of the intersection is not restricted in the current

time step and vehicles that are allowed to move at the current time step have priority over vehicle v. If $z_v^{\text{veh}} = 1$, then the constraint becomes $t_v(\gamma_v^+) + \tau_v(\gamma_v^+) \leq t + \Delta t$, which means the exit time of vehicle v should be in the current time interval.

$$z_v^{\text{veh}} \leq 1 - z_n^{\text{ped}} \delta_v^n \quad \forall v \in \mathcal{V}, \forall n \in \mathcal{W} \tag{5.45}$$

$$z_{v'}^{\text{veh}} \leq z_v^{\text{veh}} \quad \forall v, v' \in \mathcal{V} : \gamma_v^- = \gamma_{v'}^-, e_v < e_{v'} \tag{5.46}$$

$$t_v(\gamma_v^+) + \tau_v(\gamma_v^+) \leq t + \Delta t + (1 - z_v^{\text{veh}}) M_v \quad \forall v \in \mathcal{V} \tag{5.47}$$

$$z_v^{\text{veh}} \in \{0, 1\} \quad \forall v \in \mathcal{V} \tag{5.48}$$

$$z_n^{\text{ped}} \in \{0, 1\} \quad \forall n \in \mathcal{W} \tag{5.49}$$

Compared with model (5.8), AIM-*ped* has a smaller convex hull that contains all possible intersection control matrices because it has more constraints. If we generate a control policy based on the convex null mentioned in Section 5.3, this control policy may not be feasible in AIM. If we sum up the number of vehicles on the same turning movement to get y_{ij} and s_{ij} and create a feasible set that has the same form as the feasible set in model (5.8), feasible set of model (5.35) is smaller than model (5.8). Therefore, the largest demand that can be accommodated by model (5.8) and AIM-*ped* are different.

5.5 Numerical experiments

The numerical experiments are composed of two parts. In the first part, we test the effects of the pedestrian demand on the efficiency of the vehicle network when there is no limit for the pedestrian waiting time. In the second part, one constraint related to the maximum pedestrian waiting time is added to AIM-*ped* model to test its effects on intersection performance. For both experiments, simulations with different pedestrian and vehicle demands are conducted. A multi-layer network is used in the simulation, as shown in Figure 5.4.

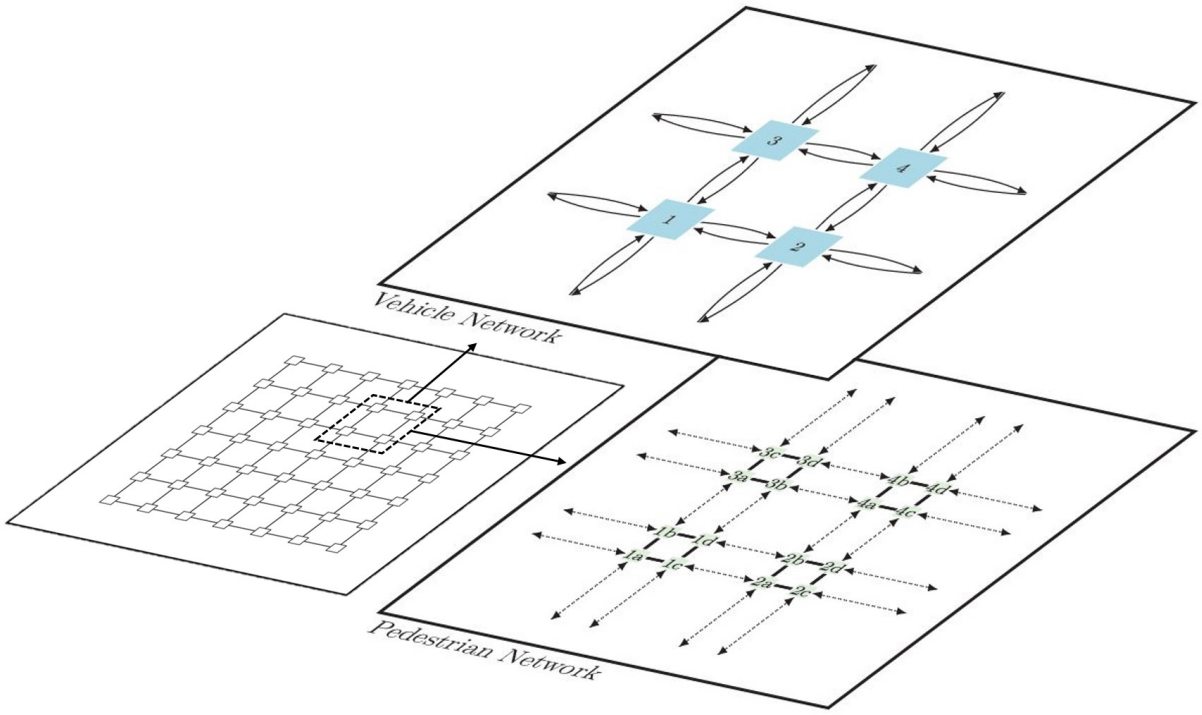


Figure 5.4: 7-by-7 grid test network

5.5.1 Vehicle network

The first layer is the vehicle layer consisting of a 7-by-7 grid of intersections. Any two adjacent intersections are connected by a pair of directed links that are 1800 feet long. These directed links represent a two-way road that connects two adjacent intersections. What is not depicted is that each of these directed links is segmented into 4 intermediate links of equal distance (450 feet). These intermediate links are required for the assumption that each link has a free flow travel time of 1 time step. Throughout the edge of the grid reside vehicle centroids, denoting either a vehicle origin or destination, adjacent to every intersection on the edge of the grid.

5.5.2 Pedestrian network

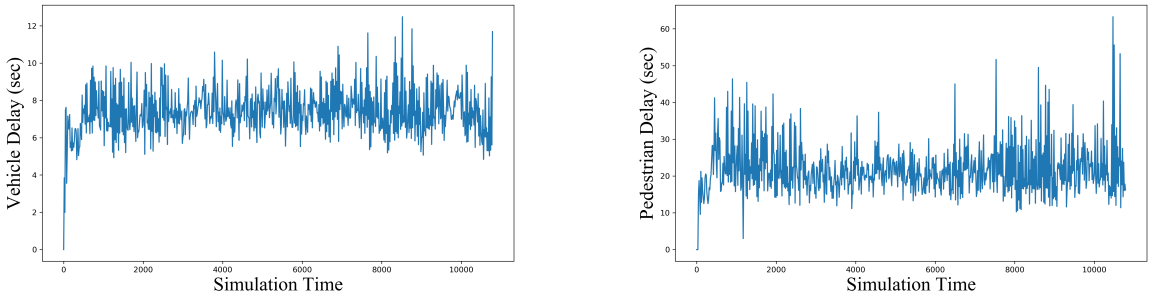
The second layer of the network is the pedestrian layer consisting of sidewalks, crosswalks, and pedestrian nodes as depicted in Figure 5.4. The second layer overlaps the vehicle layer. For every vehicle intersection there resides four pedestrian nodes, each of which is connected by a crosswalk. In Figure 5.4, vehicle node 1 and pedestrian nodes $1a, 1b, 1c,$ and $1d$ correspond to the same intersection. The solid lines connecting each pair of these four pedestrian nodes are crosswalks. Each crosswalk is controlled by a signal, which if activated, all pedestrians who are waiting on this crosswalk can move, and any vehicles who have turning movements that conflict with this crosswalk cannot move. Crosswalks are either active or inactive for an entire timestep of 15 seconds. Any two adjacent groups of four pedestrian nodes are connected by a pair of undirected links that are 1800 feet and overlay the directed vehicle links that connect adjacent intersections. These two undirected pedestrian links represent two sidewalks on both sides of a road. Pedestrians can move in both directions on either sidewalk, but cannot cross the road from one sidewalk to another. A pedestrian can only switch links, or sidewalks, by reaching the end of its current link and either traversing a crosswalk or entering another link directly that is not blocked by a

crosswalk (jaywalking is not modeled). Like vehicle links, pedestrian links are also segmented into intermediate links. Pedestrian links are made up of 40 intermediate links of equal distance (45 feet). We assume that a pedestrian will be able to traverse 45 feet in a timestep of 15 seconds at a speed of 2 miles per hour. At the corner of every intersection is a pedestrian centroid, denoting either a pedestrian origin or destination.

5.5.3 Simulation parameters

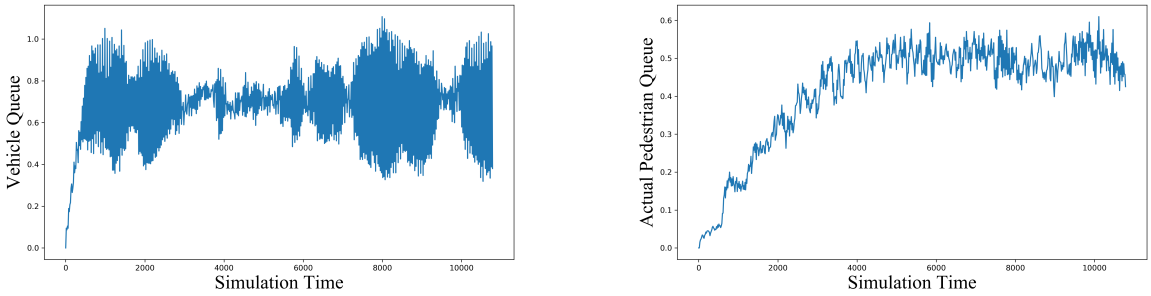
The simulation time is 3 hours with a timestep of 15 seconds. Vehicles and pedestrians continually enter the network at varying demand rates. The vehicle demand ranges from 2 to 18 vehicles per hour per origin-destination pair and the pedestrian demand ranges from 0 to 10 pedestrians per hour per origin-destination pair. Upon creation, each vehicle and pedestrian has a specified origin centroid and a specified destination centroid. For every vehicle, a random shortest path is generated and followed until the vehicle reaches its destination. For every pedestrian, a shortest path is selected from all predetermined paths and followed until the pedestrian reaches its destination. Every pedestrian origin-destination pair has 5 predetermined paths and when a pedestrian is generated, it will randomly pick one path from the 5 possible. Due to the large number of possible pedestrian paths, and the typical lack of congestion on sidewalks, we restrict the set of possible pedestrian paths to 5 per origin-destination.

Two sets of simulations are run. One set is run using estimated pedestrian queue lengths $\hat{x}_{ij}^{\text{ped}}$ as described earlier in the paper, and results are shown in Section 5.5.4. Another set is run using actual pedestrian queue lengths with x_{ij}^{ped} equal to the actual number of pedestrians waiting for the specific turning movement from i to j, and results are shown in Section 5.5.5. For every time step, mixed-integer linear programs are created and solved for each of the intersections using CPLEX. Simulations are run on a computer with Intel i7-8700 CPU and 16 GB RAM. Computation times are analyzed in Section 5.5.6 and the



(a) Vehicle delay

(b) Pedestrian delay



(c) Vehicle queue length

(d) Pedestrian queue length

Figure 5.5: Simulation results with the vehicle demand of 10 vehicles per hour per O-D pair and the pedestrian demand of 4 pedestrians per hour per O-D pair

conflict rate is analyzed in Section 5.5.7.

5.5.4 Results with estimated pedestrian queue lengths

The simulation is run over a time period of three hours and vehicle delay, pedestrian delay, vehicle queue length, and pedestrian queue length are all bounded. Vehicle delay, pedestrian delay, and vehicle queue length become stable after around 500 seconds, or about 8 minutes into the simulation, as shown in Figure 5.5(a), (b), and (c). Pedestrian queue length becomes stable after 4000 seconds, or 67 minutes into the simulation, as shown in Figure 5.5(d).

In Figure 5.6(a), we can see that an increase in pedestrian demand results in an increase in average vehicle delay. This behavior is expected because a high relative pedestrian de-

mand would result in a higher priority for pedestrians to move at an intersection and thus a higher delay for vehicles. An increase in vehicle demand has more of a mixed effect on average vehicle delay. When vehicle demand is low, increasing vehicle demand decreases average vehicle delay, and when vehicle demand is high, increasing vehicle demand increases average vehicle delay. This could be because of a balancing act between two main factors that drive vehicle delay at an intersection: queue length and vehicle weights. When vehicle demand is low, queue lengths are low and as a result vehicle weights are low. Vehicles will be given low priority compared to pedestrians and vehicle delay will be higher. However, when vehicle demand is high, queue lengths will tend to be higher and so will vehicle weights but since only a certain number of vehicles can move in a timestep there could be situations where vehicles are given high priority to move but will still have to wait for multiple timesteps because queues are long, resulting in higher delay.

Figure 5.6(b) describes the relationship between average pedestrian delay, vehicle demand, and pedestrian demand. As a trend average pedestrian delay increases as vehicle demand increases. This is explained by the fact that more vehicles at an intersection give lower priority to pedestrians, increasing the delay. Figure 5.6(b) also implies that as pedestrian demand increases, pedestrian delay decreases, regardless of whether or not pedestrian demand is high or low. This is because if a crosswalk activates, all the pedestrians waiting on that crosswalk can move. As a result, there's no conflict between pedestrian queue lengths and pedestrian weights as there is with vehicles.

Figure 5.6(c) shows the trend that the average vehicle queue length increases with the vehicle demand. The effect of pedestrian demand on vehicle queue length is not significant. We would expect that a higher pedestrian demand would increase vehicle queue length, since more pedestrians would result in higher priority for pedestrians at intersections and thus a higher likelihood that vehicles will not move and a vehicle queue will accumulate. However, pedestrian demand seems to have a minimal effect on vehicle demand in Figure 5.6(c).

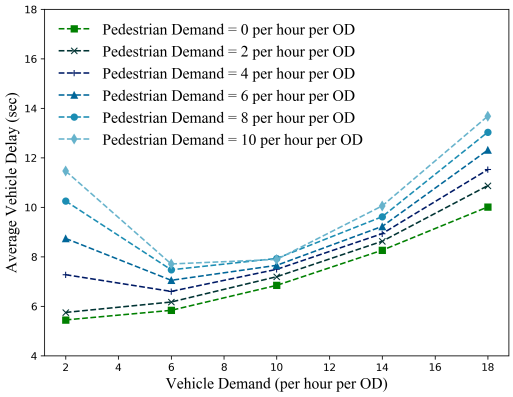
Figure 5.6(d) describes the relationship between average pedestrian queue length, vehicle demand, and pedestrian demand. Increasing pedestrian demand increases average pedestrian queue length, and increasing vehicle demand has a similar effect.

Figure 5.6(f) describes the relation between the difference between actual and estimated pedestrian queue lengths, vehicle demand, and pedestrian demand. The difference between actual and estimated pedestrian queue lengths increases as pedestrian demand increases. This is because in our simulation using estimated pedestrian queue lengths, the estimated pedestrian queue length for every turning movement is constant. However, for higher pedestrian demand, once a crosswalk is activated, the flow of pedestrians to downstream links will be much higher than the constant estimated pedestrian queue length.

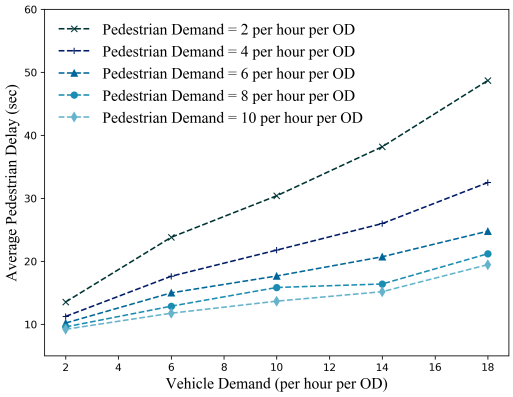
5.5.5 Results with actual pedestrian queue lengths

In Figure 5.7(a), the effect of pedestrian demand and vehicle demand on the vehicle delay is similar to that in Section 5.5.4. An increase in the pedestrian demand results in larger vehicle delay. An increase in the vehicle demand reduces the vehicle delay when the vehicle demand is small but increases the vehicle delay when the vehicle demand is large. When the vehicle demand is between 10 and 14 vehicles per hour per OD pair, the change in the vehicle demand does not significantly affects the vehicle delay. Compared with the vehicle delay in Section 5.5.4, the average vehicle delay here is larger because the estimated pedestrian queue length is lower than the actual queue length which gives higher priority to pedestrians in the intersection control.

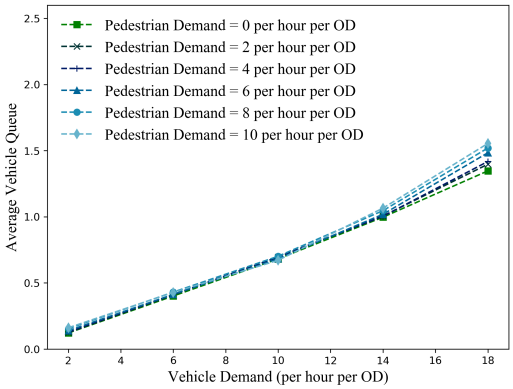
In Figure 5.7(b), an increase in the pedestrian demand reduces the pedestrian delay and an increase in vehicle demand increases the pedestrian delay. Pedestrian delay is on average higher in Figure 5.6 than in Figure 5.7. This is because our estimated pedestrian queue length for an intersection is lower than the actual pedestrian queue length, as shown in Figure 5.6(f).



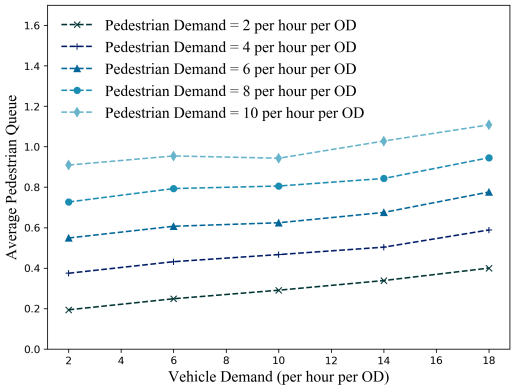
(a) Vehicle delay



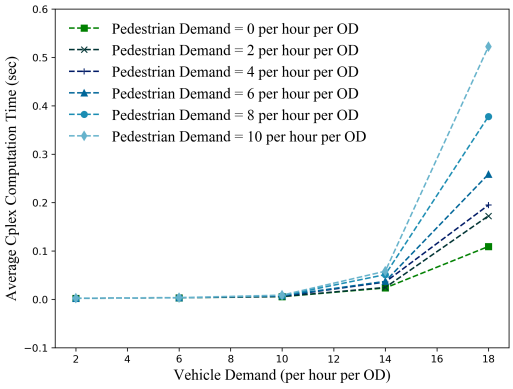
(b) Pedestrian delay



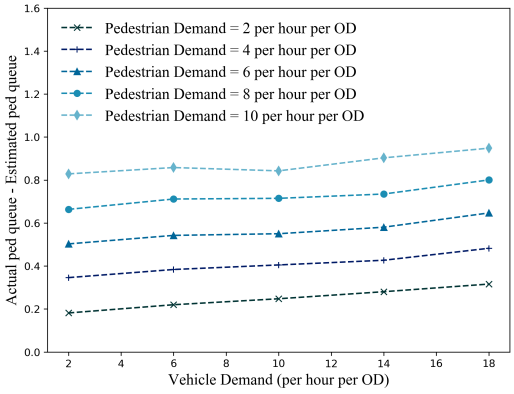
(c) Vehicle queue length



(d) Pedestrian queue length



(e) CPLEX calculation time



(f) The difference between the actual and the estimated pedestrian queue lengths

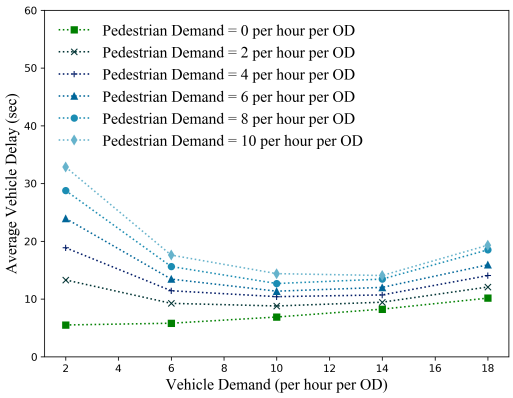
Figure 5.6: Simulation result using estimated pedestrian queue length

In Figure 5.7(c), increases in the pedestrian demand and the vehicle demand both increase the vehicle queue length. Compared with Figure 5.6(c), the effects of pedestrian demand on the vehicle delay is more significant in Figure 5.7(c).

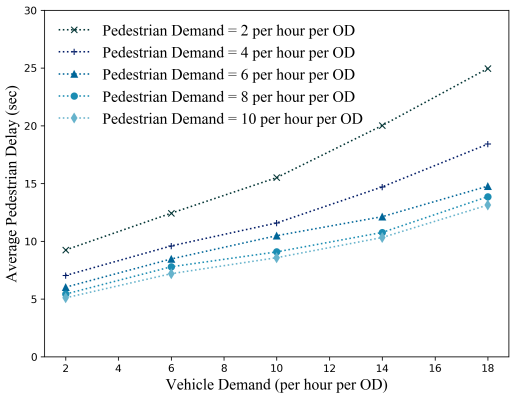
Figure 5.7(d) describes the relationship between average pedestrian queue length, vehicle demand, and pedestrian demand. An increase in the pedestrian demand results in an increase in the pedestrian delay. An increase in the vehicle demand also increases the pedestrian delay but the effect is not significant.

5.5.6 Discussion of computation times

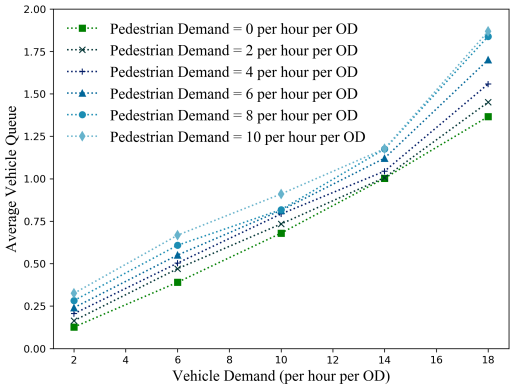
Figure 5.6(e) and Figure 5.7(e) describe the relation between CPLEX computation time, vehicle demand, and pedestrian demand. It is the average CPLEX computation time for one time step and one intersection in the simulation. As shown in Figure 5.6(e) and Figure 5.7(e), CPLEX computation time is close to zero when the vehicle demand is small and spikes upwards for high vehicle and pedestrian demand. This is because the high vehicle and pedestrian demands increase the vehicle queue lengths, and every vehicle creates several additional variables in the mixed-integer program that CPLEX needs to solve. Vehicles with different moving directions create different numbers of variables in the mixed-integer program which related with the number of conflict points they would pass through. As the proposed algorithm is a distributed algorithm, we only need the CPLEX computation time for an individual intersection to be smaller than the step size if we want to implement it in real-time. In the simulation, the average CPLEX computation time per intersection under all demands is far smaller than the time step size (15 seconds), which indicates that the algorithm can possibly be used in real-time.



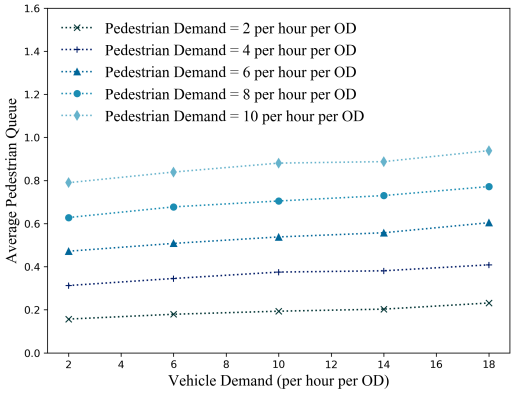
(a) Vehicle delay



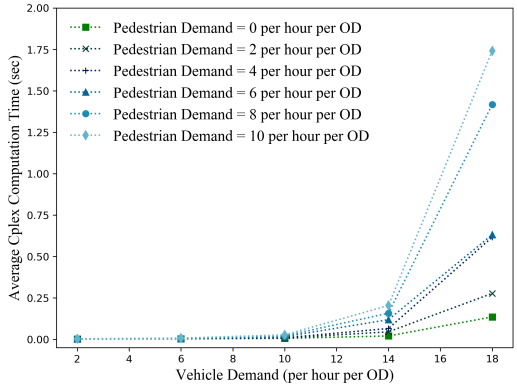
(b) Pedestrian delay



(c) Vehicle queue length



(d) Pedestrian queue length



(e) CPLEX calculation time

Figure 5.7: Simulation result using actual pedestrian queue length

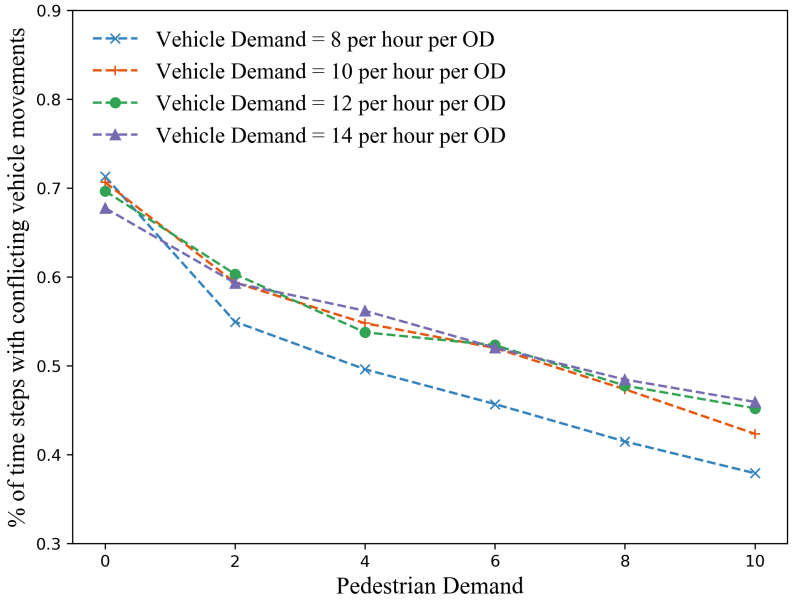


Figure 5.8: Percent of time steps that allow conflicting vehicles to move

5.5.7 The effect of pedestrian and vehicle demand on the conflict rate

Figure 5.8 describes the relationship between pedestrian demand and the similarity of the simulation to a phase-based intersection control. As pedestrian demand increases, the percentage of time steps in which conflicting vehicle movements are simultaneously permitted to traverse an intersection decreases. The simulation becomes more and more like a traditional traffic light system where only non-conflicting vehicles can move in a certain time period. This is because of the high pedestrian demand and high pedestrian priority at intersections, there is a higher likelihood that crosswalks will be activated in any given timestep which restricts all vehicle movements that go through that crosswalk. Crosswalk activation limits the different possible vehicle movements in a time step making it more likely that moving vehicles have non-conflicting trajectories. Even when the pedestrian demand is high, the algorithm still allows about 40% of the conflicting vehicle movements, which means this algorithm can adapt to pedestrian demand and activate conflicting movements when optimal.

5.5.8 Effects of pedestrian waiting time constraints

When pedestrians have been waiting for a long time at the crosswalk, they have a high probability of jaywalking. According to a study that analyzed data collected from a signalized intersection in Kunming, China, pedestrians could wait for 48.2 seconds on average before they jaywalk (Li, 2013). At some intersections, pedestrians arrive in groups. With a large number of people in a group, the waiting time of pedestrians is smaller (Hamed, 2001). Jaywalking pedestrians may force autonomous vehicles to stop and reduce the intersection efficiency. A severe traffic accident will occur if the detectors of an autonomous vehicle fail to detect jaywalking pedestrians or an autonomous vehicle detects pedestrians but operates in a way that prioritizes the safety of passengers in the car rather than pedestrians. Therefore, it is meaningful to set a limit for pedestrian waiting time and test its effects on intersection efficiency.

A new parameter is introduced to represent the waiting time limit for pedestrians at crosswalk n, denoted by $\bar{\tau}_n(t)$. In the new model, when the waiting time of pedestrians $\hat{\tau}_n(t)$ at crosswalk n exceeds its waiting time limit $\bar{\tau}_n(t)$, the crosswalk n will be activated. The mixed-linear program in Section 5.4 is modified, as shown in program (5.50).

$$\max \quad \sum_{v\in\mathcal{V}} w_v^{\text{veh}} z_v^{\text{veh}} + \sum_{n\in\mathcal{W}} w_n^{\text{ped}} Q_n^{\text{ped}} z_n^{\text{ped}} \tag{5.50a}$$

$$\text{Constraints } (5.36) - (5.49)$$

$$\left(1 - z_n^{\text{ped}}\right)\left(\hat{\tau}_n(t) - \bar{\tau}_n(t)\right) \geq 0 \tag{5.50b}$$

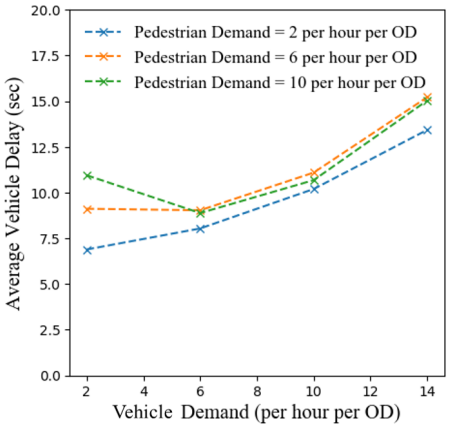
An additional constraint for the limit of pedestrian waiting time is added while the objective function and other constraints in Section 5.4 keep the same. In constraint (5.50b), when $\hat{\tau}_n(t) \leq \bar{\tau}_n$, the value of $S_n(t) \in \{0, 1\}$ can be either 0 or 1, to make the left-hand side less than or equal to 0. When $\hat{\tau}_n(t) > \bar{\tau}_n$, $S_m(t) \in \{0, 1\}$ has to be 1, which means crosswalk n should be activated. Constraint (5.50b) may reduce the efficiency of vehicles

at the intersection.

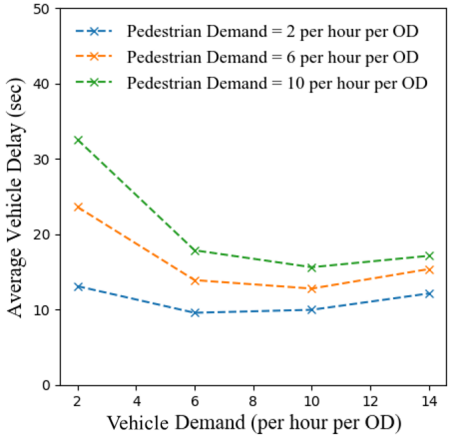
The demand rates for pedestrians are set to be 2, 6, and 10 per hour per OD pair, and the demand rates for vehicles are set to be 2, 6, 10, and 14 per hour per OD pair. The waiting time limit is set to be 30, 120, and 240 seconds. The following results show the effects of the waiting time limit on vehicle delays and pedestrian delays.

Figure 5.9 shows vehicle delays under different demands and waiting time limits. The effect of vehicle demand on vehicle delays shows a similar trend as that in Figure 5.6a and Figure 5.7a. As the vehicle demand increases, the vehicle delay first decreases and then increases. The decrease in the delay results from the increase of the weight of vehicle queues when the queue length gets longer. When the queue length is too long to be clear in one time step as the demand further increases, the delay gets larger again. An increase in pedestrian demands leads to an increase in vehicle delays. Comparing Figure 5.9a, Figure 5.9d, and Figure 5.9e, we find the effect of the waiting time limit on vehicle delays. When the demand is small, the waiting time limit has nearly no effect on delays because all queues can be cleared in a short time. When the demand is large, under a waiting time limit of 30 seconds, the average waiting time is 15 seconds, which is 5 seconds larger than the scenario when the waiting time limits of 120, and 240 seconds are used. Comparing Figure 5.9a, Figure 5.9d, and Figure 5.9e, we can find that for the scenario using actual pedestrian queue measurements, the waiting time limit has little effects on vehicle delays. Figure 5.10 plots the difference between scenarios with and without the waiting time limit under the same demand. A positive value indicates that the waiting time limit increases the vehicle delay. In Figure 5.10, when vehicle demand is large, the waiting limit increases the vehicle delay by as large as 6 seconds and 3 seconds per vehicle when the estimated pedestrian queue and measured pedestrian queue are used. When the demand is small, the waiting time limit has little effect.

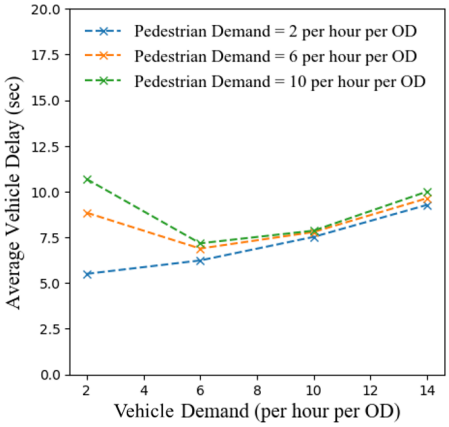
Figure 5.9 shows pedestrian delays under different demands and waiting time limits. The pedestrian delay increases with both vehicle demand and pedestrian demand. A smaller



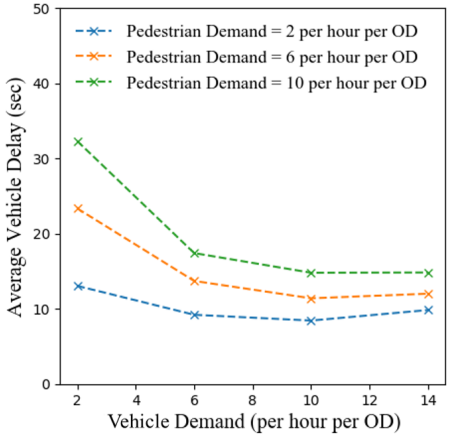
(a) threshold = 30 sec (estimated queue)



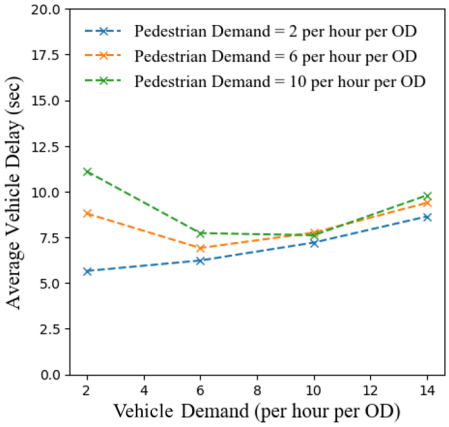
(b) threshold = 30 sec (actual queue)



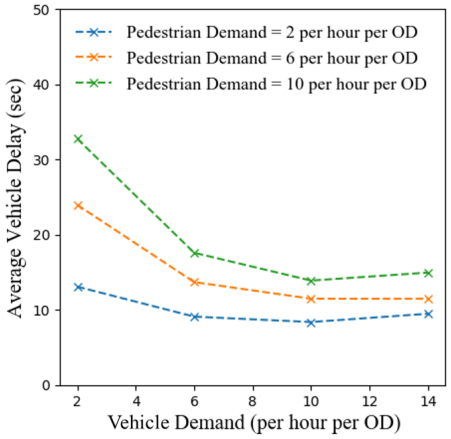
(c) threshold = 120 sec (estimated queue)



(d) threshold = 120 sec (actual queue)

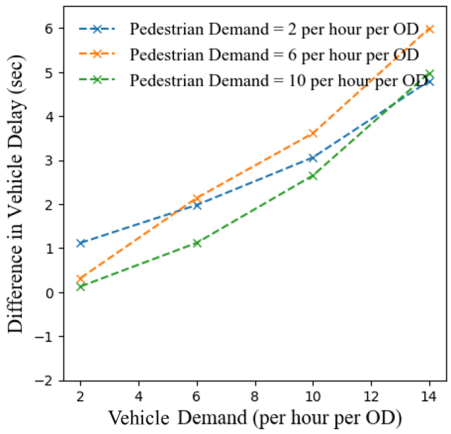


(e) threshold = 240 sec (estimated queue)

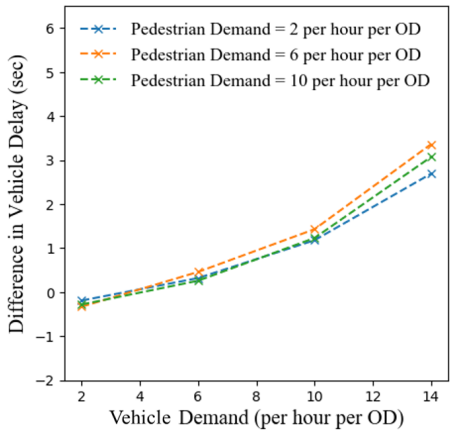


(f) threshold = 240 sec (actual queue)

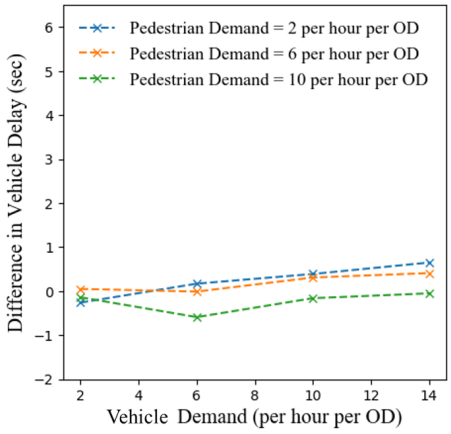
Figure 5.9: Vehicle delays in simulation with estimated and actual pedestrian queues



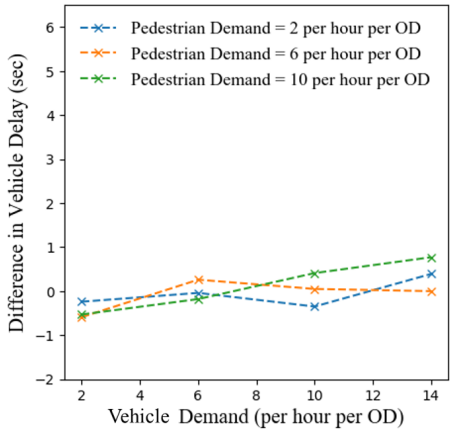
(a) time limit = 30 sec (estimated queue)



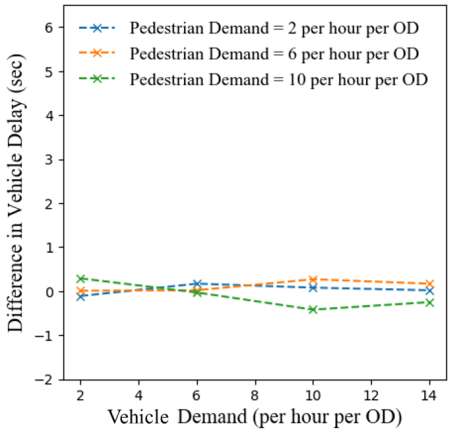
(b) time limit = 30 sec (actual queue)



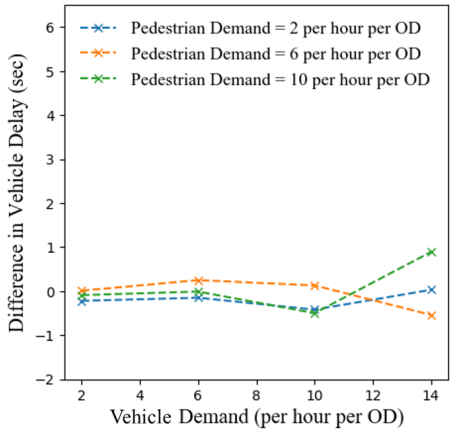
(c) time limit = 120 sec (estimated queue)



(d) time limit = 120 sec (actual queue)

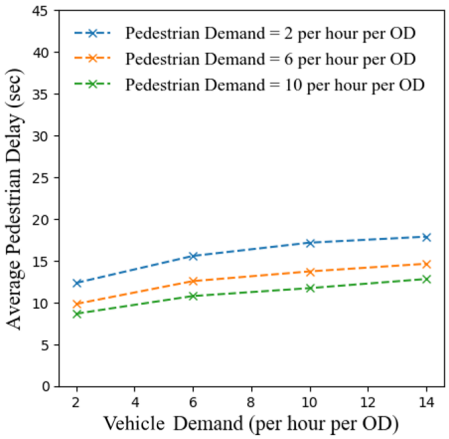


(e) time limit = 240 sec (estimated queue)

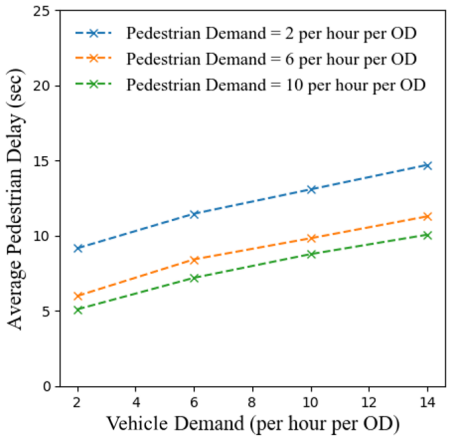


(f) time limit = 240 sec (actual queue)

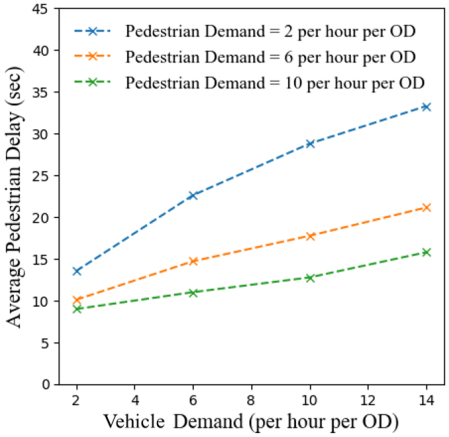
Figure 5.10: The difference in vehicle delays with estimated and actual pedestrian queues



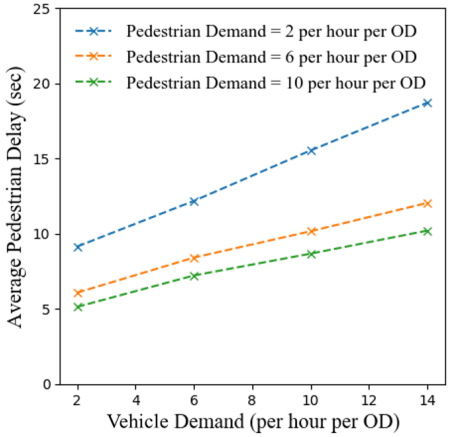
(a) time limit = 30 sec (estimated queue)



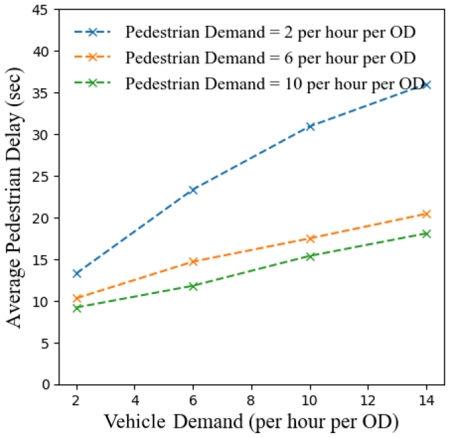
(b) time limit = 30 sec (actual queue)



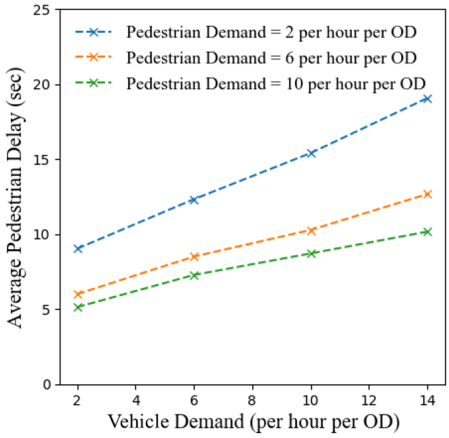
(c) time limit = 120 sec (estimated queue)



(d) time limit = 120 sec (actual queue)

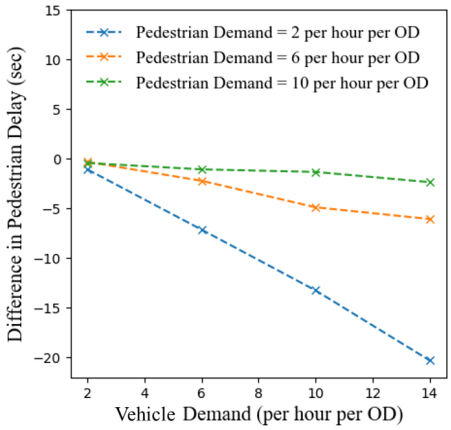


(e) time limit = 240 sec (estimated queue)

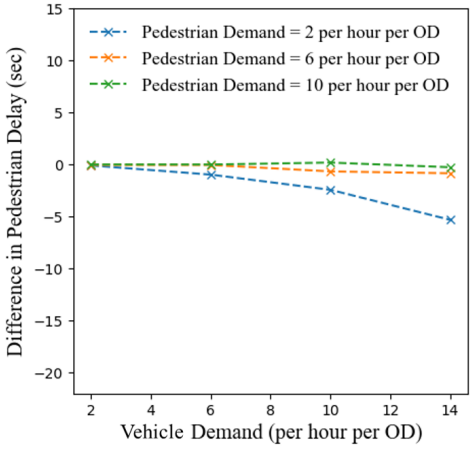


(f) time limit = 240 sec (actual queue)

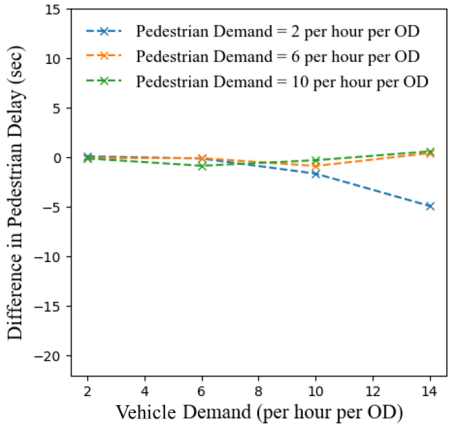
Figure 5.11: Pedestrian delays in simulation with estimated and actual pedestrian queues



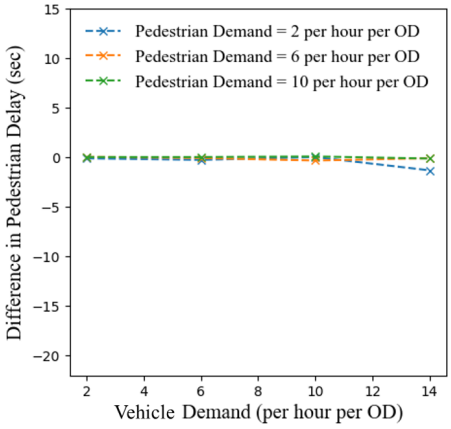
(a) time limit = 30 sec (estimated queue)



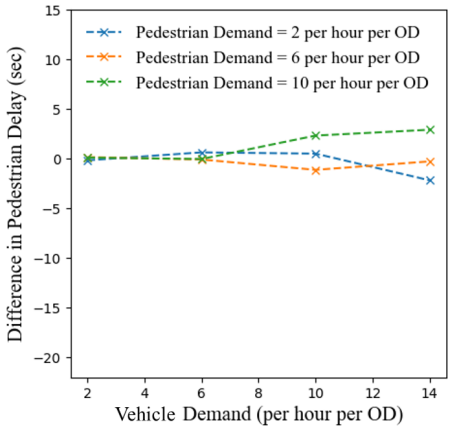
(b) time limit = 30 sec (actual queue)



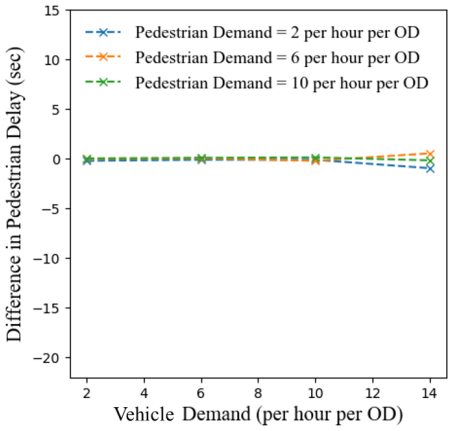
(c) time limit = 120 sec (estimated queue)



(d) time limit = 120 sec (actual queue)



(e) time limit = 240 sec (estimated queue)



(f) time limit = 240 sec (actual queue)

Figure 5.12: The difference in vehicle delays with estimated and actual pedestrian queues

waiting time limit helps to reduce pedestrian delays. In Figure 5.12, when the pedestrian demand is small and the vehicle demand is large, the waiting limit has greater effects on reducing pedestrian delays. In the scenarios using estimated pedestrian queue length, when the pedestrian demand is 2 pedestrians per hour per OD pair, the vehicle demand is 14 vehicles per hour per OD pair, and the waiting time limit is 30 seconds, the reduction in pedestrian waiting time is 20 seconds. For the corresponding scenario using the actual pedestrian queue length, the reduction is 5 seconds. When the vehicle demand is much larger than the pedestrian demand, the length and the weight of pedestrian queues are small, which results in a large waiting time. A small waiting time limit can greatly solve this problem in this condition.

5.6 Conclusions

The chapter proposes an autonomous intersection management algorithm based on max-pressure control considering both vehicles and pedestrians. This chapter defines the stability region of the traffic demand and proves that this algorithm can produce throughput-optimal intersection control at the network level. To apply this algorithm in simulation, this chapter combines an existing trajectory optimizing algorithm with max-pressure control and formulates a mixed-integer program model to calculate the optimal trajectories of vehicles and optimal control of pedestrian signals based on the max-pressure control. In the simulation, the proposed algorithm has a small computation time when the vehicle demand is small. Simulation results show that the pedestrian and vehicle delays become stable in a short time and the difference between the actual and the estimated pedestrian queue lengths are bounded. Simulation results with different demands of pedestrians and vehicles show the trade-off between the efficiency of vehicles and pedestrians in this algorithm. Delays of pedestrians and vehicles are negatively correlated. When demands of vehicles and pedestrians increase, the intersection control produced by this algorithm is more similar to

phase-based intersection control. When a constraint limiting the waiting time of pedestrian queues is added to the model, the waiting time of pedestrians is greatly reduced when the vehicle demand is much larger than the pedestrian demand. Besides, with large demand, this algorithm is still capable of adapting to the change of the pedestrian demand and allows vehicles with conflicting trajectories to pass the intersection.

Chapter 6

User Equilibrium Analysis with Max-Pressure Control

When designing and testing the effect of the max-pressure control in traffic networks, most studies assume constant average turning proportions from one link to another link. This assumption may not be realistic as road users react to the intersection control by selecting routes with shorter travel times, which are composed of link travel times and intersections waiting times. When max-pressure control is used in practice, the route choice behavior of road users may offset its effectiveness. This chapter designs the process to test the effects of route choice to the performance of max-pressure control using traffic assignment.

6.1 Network model and equilibrium

This chapter uses the same network model introduced in Chapter 3, including the traffic flow model, stability region, and the weight function. However, to mathematically define the network equilibrium, there are other variables that need to be clarified.

Let $\Pi_{rs} = \{\pi_{rs}^k, k = 1, ..., m\}$ be a set of paths connecting the origin r and the destination s. Let $c_\pi(t) = \min_{\pi \in \Pi_{rs}} \{c_\pi\}$ be the travel cost of path π for road users at time interval t and T is the time horizon. The path cost is mainly composed of travel cost on links and waiting cost at intersections, but these two variables are difficult to be formulated in

dynamic traffic assignment. $c^{\star}_{rs}(t)$ is the minimum travel cost connecting origin r and destination s at time interval t. The path flow of path π at time t is denoted by $h^{\pi}(t)$. Let H be the set including all path flows at all time intervals. Let $\delta^{\pi}_{i} \in \{0, 1\}$ indicate whether path π uses link i. Then the movement flow f_{ij} is related to path flow h^{π} with:

$$f_{ij} = \frac{1}{|T|} \sum_{t \in T} \sum_{\pi \in \Pi} h^{\pi}(t) \delta^{\pi}_{i} \delta^{\pi}_{j} \tag{6.1}$$

Path flows also determine turning proportions:

$$\bar{p}_{ij} = \frac{\sum\limits_{t \in T} \sum\limits_{\pi \in \Pi} h^{\pi}(t) \delta^{\pi}_{i} \delta^{\pi}_{j}}{\sum\limits_{t \in T} \sum\limits_{\pi \in \Pi} h^{\pi}(t) \delta^{\pi}_{i}} \tag{6.2}$$

which is the conditional probability that a vehicle will use link j given that it uses link i. The entering flow on link i is calculated using the summation of the product of path flows and the indicator δ^{π}_{i}.

$$d_{i}(t) = \sum_{\pi \in \Pi} h^{\pi}(t) \delta^{\pi}_{i} \tag{6.3}$$

In Chapter 3, the average turning proportion $\bar{\mathbf{p}}$, the average demand on entry links $\bar{\mathbf{d}}$ determine the average flow f_{ij} and further determine the stability region. When the route choice of road users is considered, the average turning proportion $\bar{\mathbf{p}}$ and the average demand on entry links $\bar{\mathbf{d}}$ can change with the path assignment H as the route choices of road users are affected by the flow distribution, so the stability region of the network can also be affected by the path assignment H.

We define user equilibrium as usual: vehicles choose minimum-travel time routes. Equivalently, that can be written as

$$h^{\pi}(t)(c_{\pi}(t) - c^{\star}_{rs}(t)) = 0 \tag{6.4}$$

In equation (6.4), all used paths connecting origin r and destination s at time t have travel times that equal the shortest path travel time.

6.2 Simulation-based dynamic traffic assignment

To validate the existence of a user equilibrium in a network with the max-pressure control, simulations were used to get traffic assignment results. The simulator of this experiment was built in Java. The flow propagation of this simulator is based on the cell transmission model (Daganzo, 1994).

Figure 6.1 shows the simulation process that integrates max-pressure control and the traffic assignment. The max-pressure control needs the input of the turning proportion of each turning movement in the network. Before the first iteration of the simulation, the proportion of demand on each path is defined, based on which the turning proportion of each turning movement is calculated. To get the initial values of path probabilities, we assume road users from an origin to a destination only use the shortest path connecting this origin-destination pair with its path probability set to be 100%.

At each time step, each intersection calculates the weights of movements based on turning proportions for the current iteration and current queue lengths. Then the phase with the largest weight is activated. The intersection control affects vehicle delays at intersections and further affects the total travel time and vehicle route choices. During the simulation, the average link travel time and the intersection delay are updated and recorded, based on which the shortest path information for each origin-destination pair is updated. In the following iteration, some vehicles switch their paths to the new shortest path. In this chapter, the method of successive averages is used so the proportion of vehicles changing their routes is calculated as $\frac{1}{k}$, where k is the iteration index. The change in vehicle routes leads to the change in path flows and turning proportions of turning movements and further affects weights and delays of turning movements. The gap is calculated at the end of each iteration

to determine if the simulation should stop. The threshold is set to be 1%. After iterations, all used paths for each origin-destination pair will have similar travel times with negligible differences, which indicates that the traffic assignment achieves the user equilibrium. If the iteration number exceeds 100, we consider this network cannot converge.

The gap $\mathcal{G}$ is defined as the relative difference between the total system travel time $TSTT$ and the total shortest path travel time $SPTT$. In equation (6.5), $\mathcal{V}$ is a set for vehicles and ν is one vehicle. ATT_ν and STT_ν are the actual travel time of vehicle ν and the shortest path travel time between the origin and the destination pair of vehicle ν respectively.

$$TSTT = \sum_{\nu \in \mathcal{V}} ATT_\nu \tag{6.5}$$

$$SPTT = \sum_{\nu \in \mathcal{V}} STT_\nu \tag{6.6}$$

$$\mathcal{G} = \frac{TSTT - SPTT}{TSTT} \times (100\%) \tag{6.7}$$

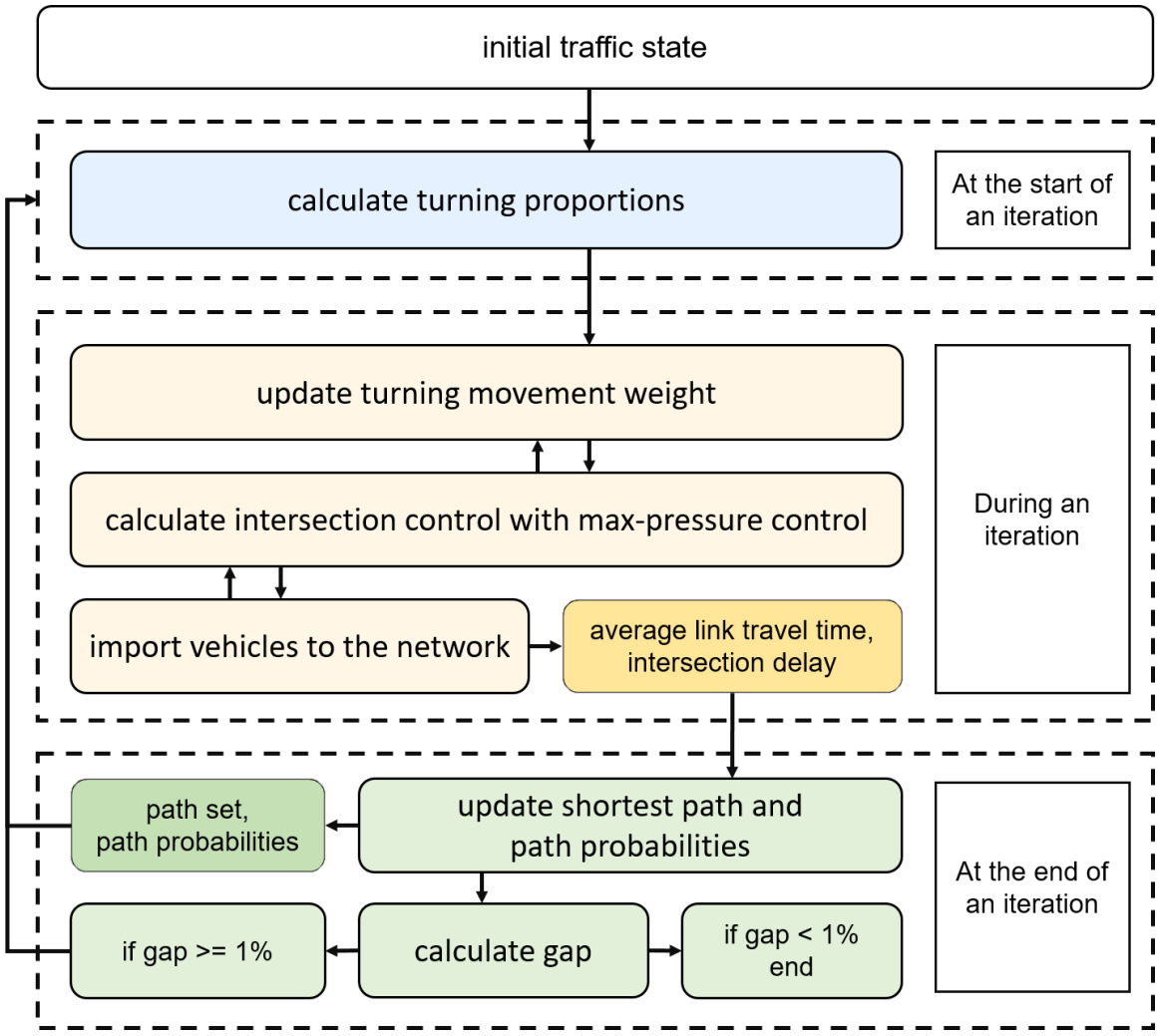


Figure 6.1: Simulation process with the max-pressure control

6.3 Numerical experiments

In the numerical experiments, 25 values of demand are tested, ranging from 8000 to 200000 vehicles in a 4-hour period. A median-size network is used, whose information is shown in Table 6.1. To compare with max-pressure control, we also obtain the simulation results of traffic assignment with fixed-time signal control. The simulation process is similar to the simulation with the max-pressure control while the only difference is that the intersection control does not react to the vehicle route choice behaviors.

Table 6.1: Test Network Information

Network	Zones	Intersections	Link	Demand	Demand Duration	Scenarios
Downtown Austin	171	546	1237	62836	2 hours	12

In this section, we discuss the existence of user equilibrium, the effects of route choices on queue length stability, traffic efficiency, and flow distribution, and the relation between stability and model convergence.

6.3.1 Existence of the user equilibrium

When the gap is smaller than or equal to 1%, we consider the traffic assignment result converges. If a scenario takes more than 100 iterations to converge, we consider this scenario hard to converge or not able to converge in the end. Figure 6.2 shows the convergence of simulations under different demands. The orange line represents the iteration number each scenario needs to converge. The overall trend is that the model converges faster when the demand is small. When the network is more congested, it is harder for dynamic traffic assignment to reach the user equilibrium. All simulations with a demand of fewer than 88000 vehicles can converge in less than 10 iterations. However, this is not always the case when the demand gets larger. The simulation with a demand of 120000 vehicles converges

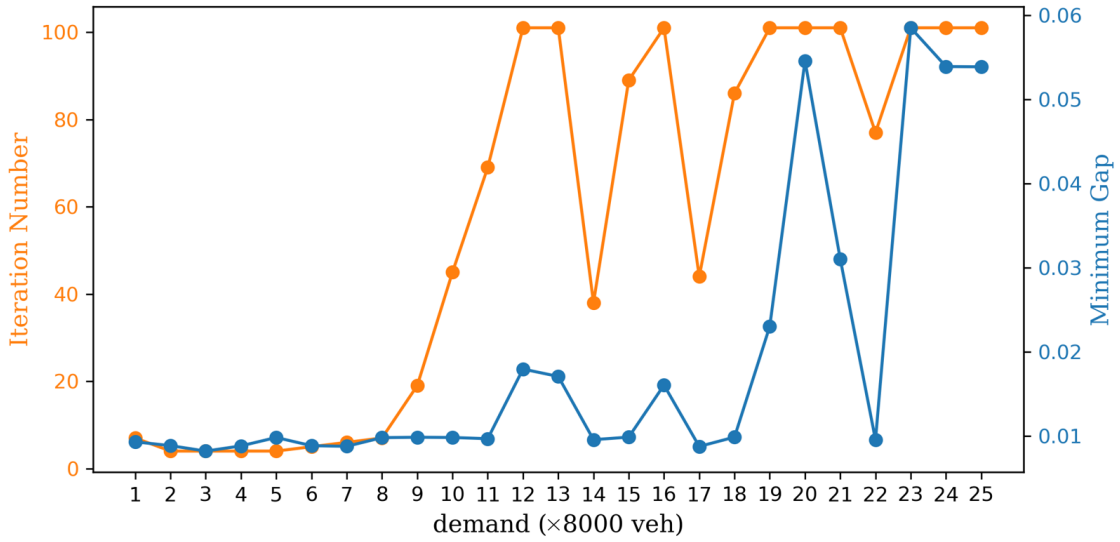


Figure 6.2: Convergence under different demands

in 89 iterations, but the simulation with a demand of 96000 vehicles needs more than 100 iterations to converge. The blue line shows the minimum gap that a scenario can achieve in 100 iterations. When the demand is larger, it is harder to achieve a small gap, which is similar to the trend of the needed iteration number to converge.

Figure 6.3 shows the change of gaps with iterations for scenarios using different demands. Four curves represent the scenarios using demands of 64000, 80000, 96000, 144000 vehicles, respectively. Only the network with a demand of 96000 vehicles does not converge, as shown in the green curve. The gap of this curve reduces as the iteration number increases, but the decrease also gets smaller. The gap in the 100th iteration is about 1.75%. For scenarios with 64000 and 80000 vehicles, they converge in less than 50 iterations. For the scenario with 144000 vehicles, the gap is always larger than 1% before the last iteration when the gap drops to a value smaller than 1%. This situation is found in other converged scenarios with large demands, such as scenarios with 144000 and 176000 vehicles, as shown in Figure 6.4. Both scenarios converge in 100 iterations, and there is a sudden drop in the gap in the last iteration for both scenarios. As the gap is calculated using TSTT

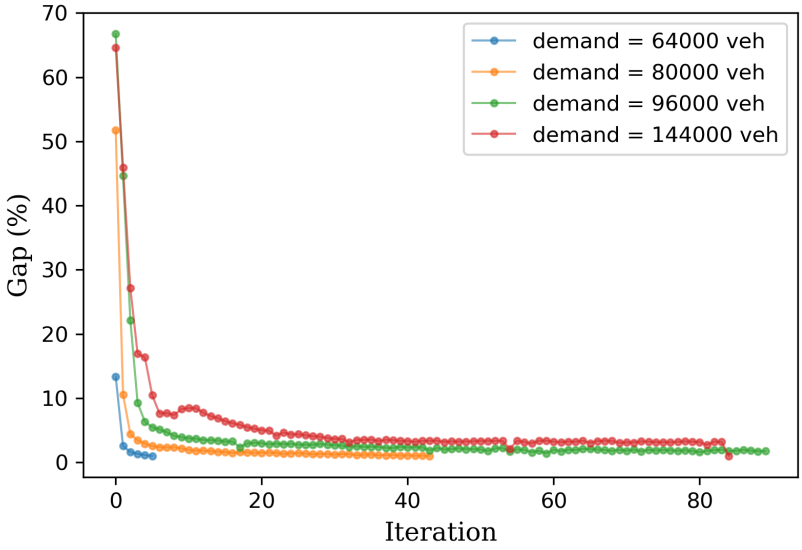


Figure 6.3: Gaps in different iterations (with cases that do not converge)

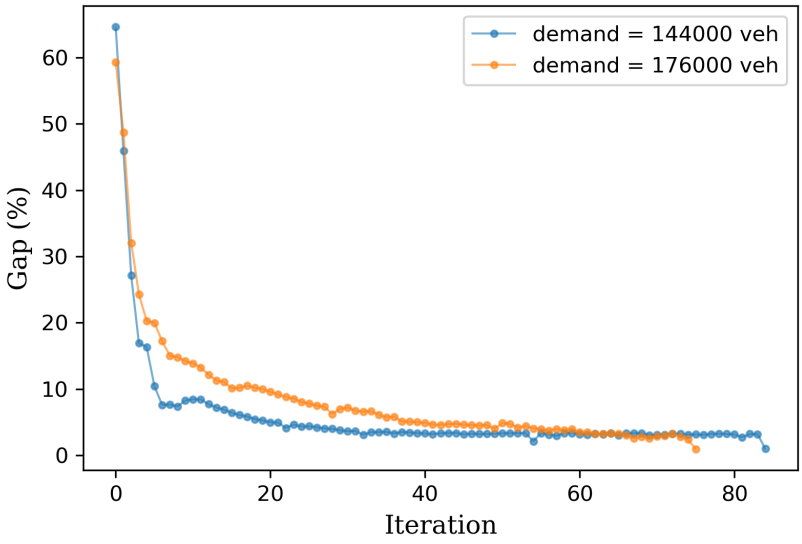


Figure 6.4: Gaps in different iterations (with two cases that converge)

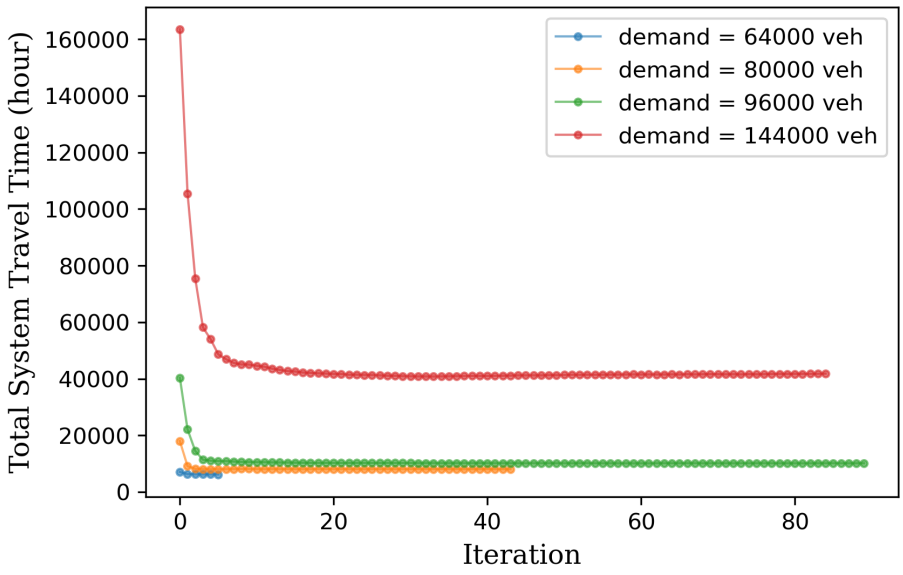
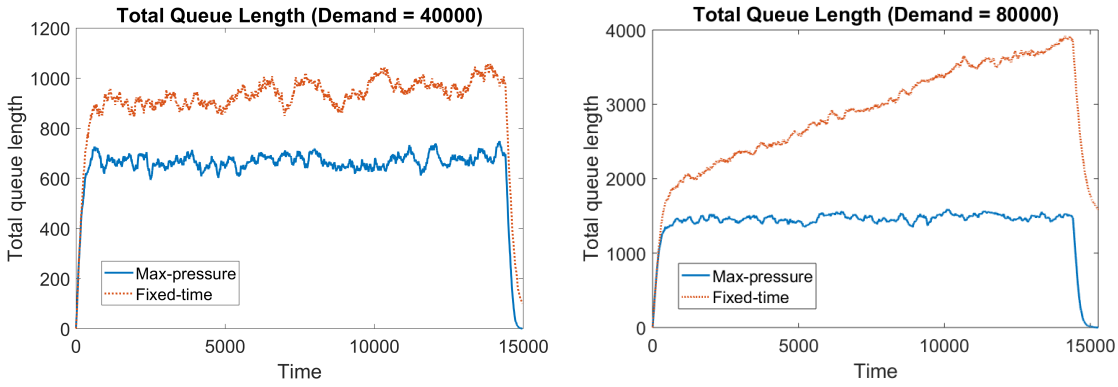


Figure 6.5: Total system travel times in different iterations

and SPTT, the sudden drop in the value may result from the change in these two variables. Figure 6.5 plots the change in TSTT in different iterations. TSTT reduces with a large rate in first few iterations, then the change becomes smaller and smaller. In the scenario with 144000 vehicles, there is no sudden change in TSTT in the last iteration. Therefore, there should be an increase in the value of SPTT in the last iteration, which results in the sudden change of the gap. A small change in path flow can result in a change in the phase activation with the max-pressure control, further affecting the path travel time.

6.3.2 Queue length stability

One property of the max-pressure control is stabilizing the queue length when the demand is feasible. We test the maximum demand that the max-pressure control and the fixed-time signal control can address. For the test network, the maximum demand that the fixed-time control can address is 40000 vehicles, while the maximum demand that the max-pressure control can address is 88000 vehicles. When the demand exceeds the maximum demand of



(a) Total queue length (demand = 40000)

(b) Total queue length (demand = 80000)

Figure 6.6: Total queue length variation with time

the control, the queue length becomes unstable.

Figure 6.6 shows the variation of the total queue length with time. Figure 6.6a uses a demand of 40000 vehicles. Under this demand, the total queue length of the network using the fixed-time control is not stable. The total queue length increases from about 900 vehicles to 1050 vehicles. For the network using the max-pressure control, the total queue length stays at 700 vehicles. The max-pressure control is able to address larger demands than the fixed-time control. Figure 6.6b uses a demand of 80000 vehicles. Under this demand, the max-pressure control can still stabilize the total queue length as the value is constant at 1400 vehicles, while the fixed-time control cannot stabilize the total queue length as the value increases from 2000 vehicles to 3900 vehicles.

Figure 6.7 shows the variation in the total queue length with time in scenarios with 40000, 80000, 128000, and 144000 vehicles. As shown in Figure 6.6, when the demand less than 80000 vehicles, the max-pressure control can stabilize the total queue length, which is consistent with the result in Figure 6.7. When the demand is not stabilized, the queue length will keep increasing. In Figure 6.7, the total queue length for the green curve and the red curve keep increasing. When the demand can not be stabilized, the traffic assignment result can still converge (see the red curve in Figure 6.3), but the probability of

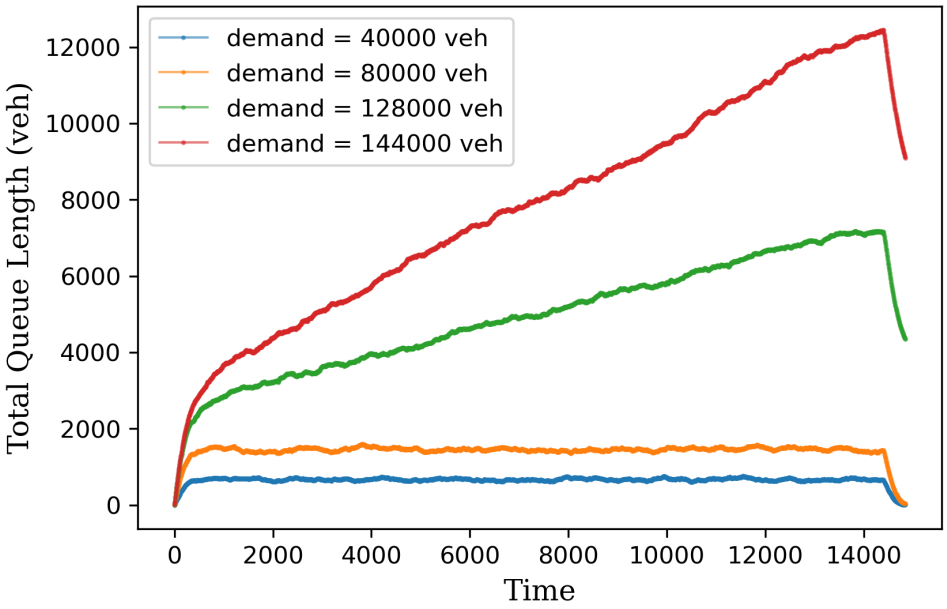


Figure 6.7: Total queue length variation under different demands

convergence gets smaller.

6.3.3 Relation between queue length stability and model convergence

It is worth exploring the relation between the queue length stability and the convergence of the traffic assignment result. Figure 6.8, Figure 6.9, and Figure 6.10 show the variation of total queue length with time in different iterations.

In Figure 6.8, the demand is small and the queue length variation has little difference among iterations. This scenario takes only 4 iterations to reach the user equilibrium, and the travel flow pattern barely changes in iterations. In the first iteration in dynamic traffic assignment, each OD pair will find the shortest path connecting them and assign all flows to the shortest path. It affects the flow distribution in the next iteration and the turning proportion of each movement, which is a variable closely related to the max-pressure control. A change in the turning proportion will result in a change in the stability region of

the network, which means a demand that can not be stabilized originally can be stabilized in the following iterations. According to Figure 6.8, the shortest path connecting the same OD does not change in four iterations.

In Figure 6.8, the queue length is not stable in the first iteration but is stable after the third iteration. After the first iteration, the flow for each OD pair is more evenly distributed on multiple paths. In the first iteration, the initial value of the turning proportion corresponds to a stability region that does not include the current value of demand, so that the controller cannot stabilize the queue length. When the demand value is close to the boundary of the current stability region, the turning proportion plays an important role in the queue length stability. It also indicates that when evaluating the performance of the max-pressure controller, it is important to use the correct values of turning proportions for movements because the controller is sensible to the turning proportion.

In Figure 6.10, the queue length is not stable and the traffic assignment result does not converge in 100 iterations. As the iteration number increases, the total queue length reduces, but the queue length still cannot be stabilized. The reduction in the total queue length is much larger for the first few iterations that the last few iterations.

Based on these figures, it can be concluded that the stability region of the demand changes with the path assignment in each iteration. For a small demand, the demand is always in the stability region, the controller can stabilize the network queue length. For a moderate demand, it is possibly included in the stability region with more iterations. When the demand is too large, there is no stability region that can include this demand even after multiple iterations. Overall, it is not recommended to use the result of all-or-nothing assignment, in which vehicles for each OD pair only use one path, to calculate the turning proportion, which produces large errors.

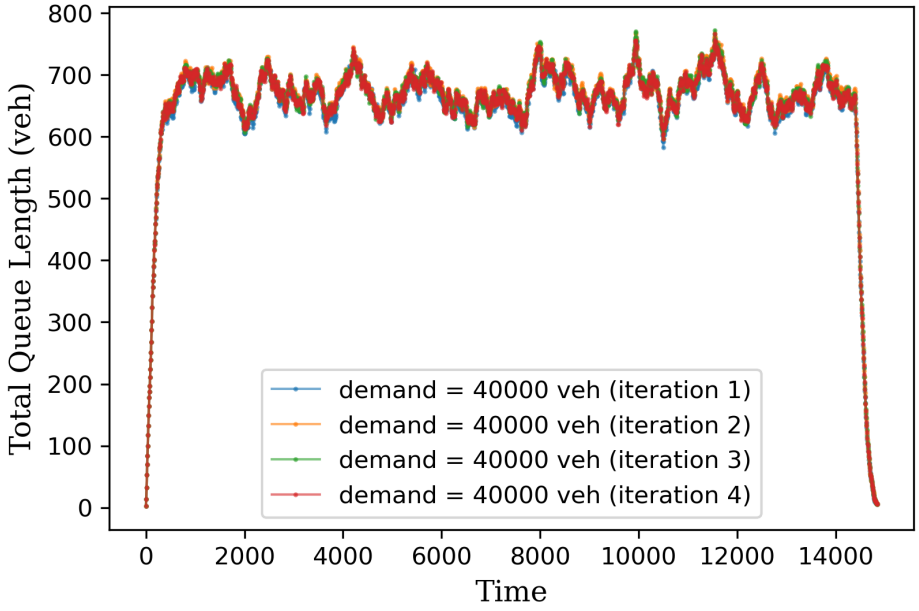


Figure 6.8: Queue length variation (always stable)

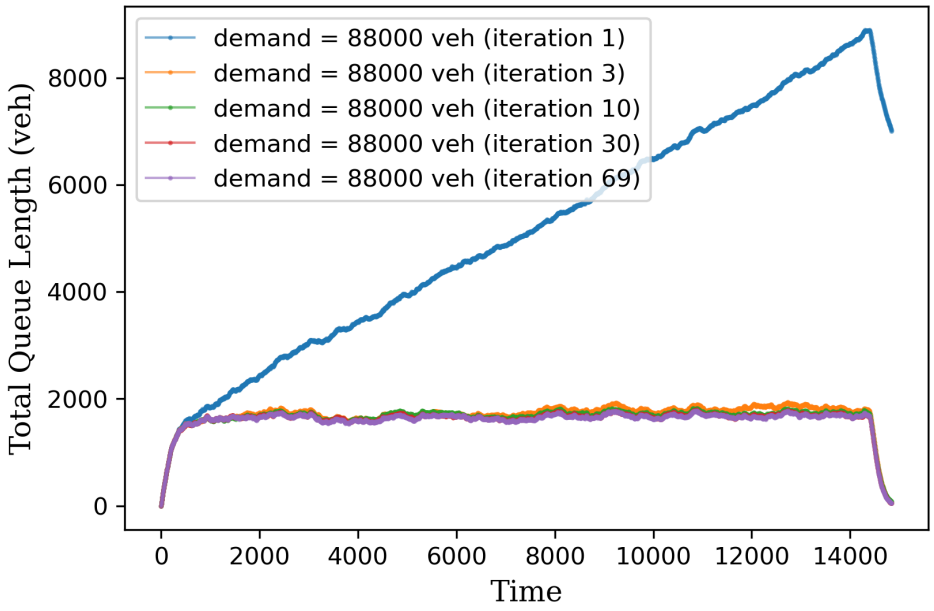


Figure 6.9: Queue length variation (can be stable)

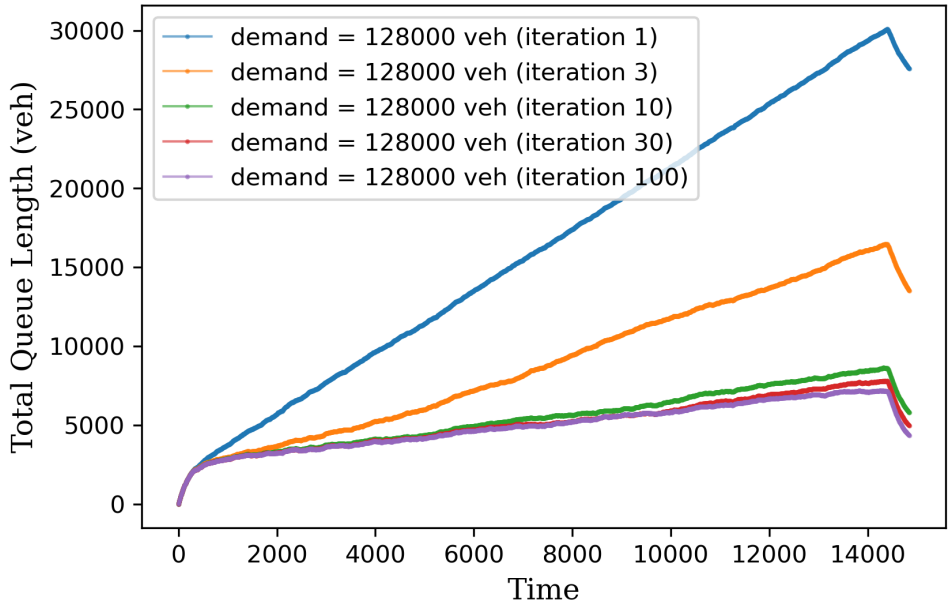


Figure 6.10: Queue length variation (cannot be stable)

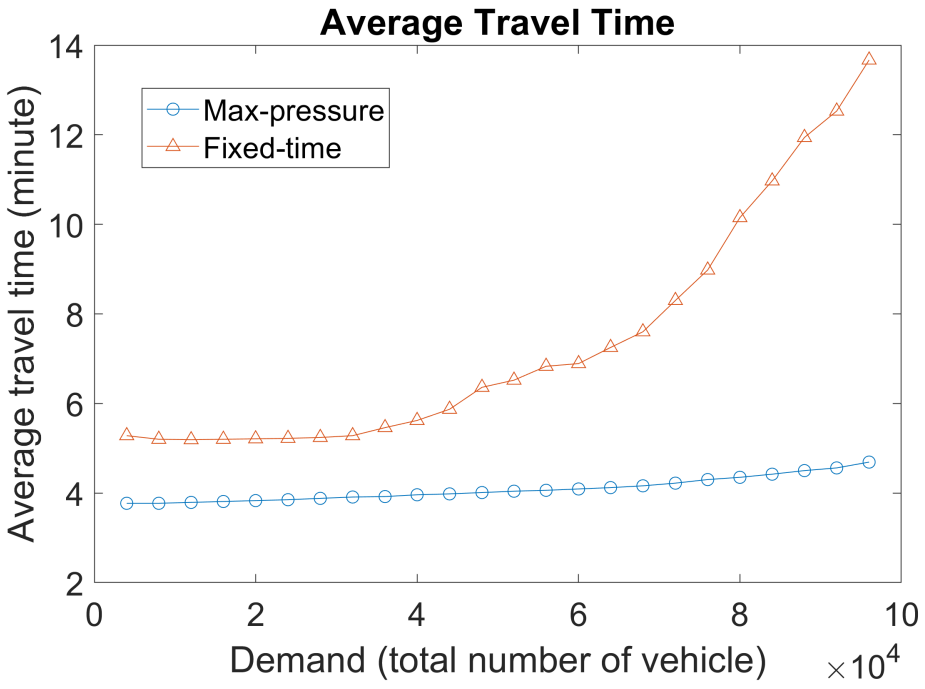


Figure 6.11: Average travel time

6.3.4 Traffic efficiency

Figure 6.11 compares the vehicle average travel times of two scenarios using the fixed-time control and the max-pressure control respectively under different demands. The average travel time of the scenario using the fixed-time intersection is much larger than that of the max-pressure control. As fixed-time signal control cannot adapt to the change of the link flow, the average delay at the intersection using the fixed-time intersection control is larger than the intersection using max-pressure control. As the demand increases, the average travel times for both scenarios increase but the scenario using the fixed-time control has a larger changing rate when the scenario using the max-pressure is only slightly affected by the demand and has a smaller changing rate.

6.3.5 Congestion distribution

In every iteration of dynamic traffic assignment, the flow is redistributed from other used paths to the shortest path, so the congestion pattern will change as well. We calculate a travel time indicator v using the link average travel time divided by the link free-flow travel time (shown in equation (6.8)) to indicate if the link is congested. A large value of v indicates a more congested link. Figure 6.12 and Figure 6.13 show the congestion distribution in different iterations under demands of 32000 and 80000 vehicles. The change in flow distribution in the first few iterations will be much larger than the change in the last few iterations. To observe the change in congestion distribution, the average link volume for all links in the first few iterations are plotted.

$$v = \frac{\text{Measured Travel Time}}{\text{Free Flow Travel Time}} \tag{6.8}$$

Figure 6.12 shows travel time indicators in the 1st, 2nd, and 4th iterations under a moderate demand. There is no significant difference between the traffic flow distribution in these

Figure 6.12: Link travel times under moderate demand

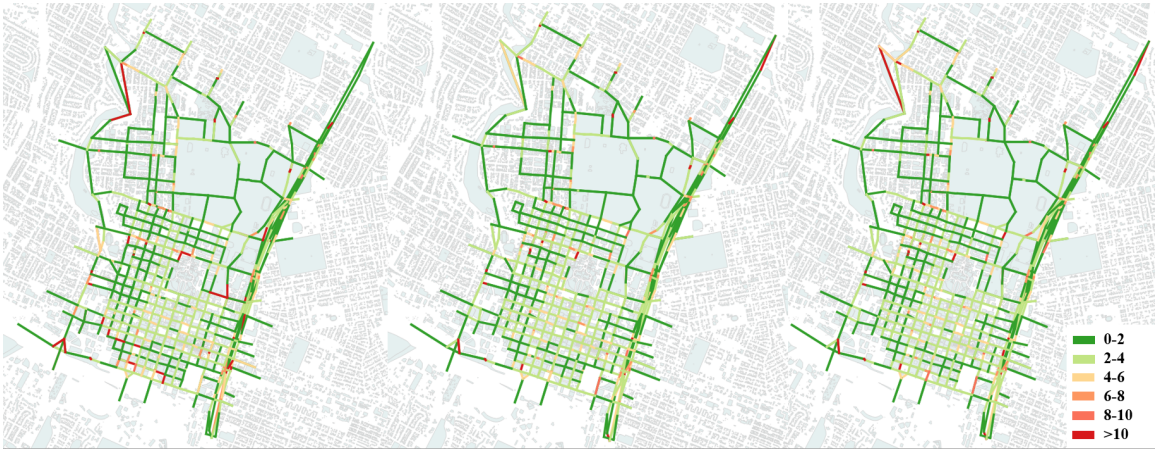


Figure 6.13: Link travel times under large demand

iterations. One noticeable difference is that the link volumes of links adjacent to intersections 5773 and 5760 get much smaller because different paths are chosen in the 2nd and the 4th iteration and new paths do not pass these two intersections. Figure 6.13 shows travel time indicators under high demand. One observed trend is that the link volumes for internal links get smaller and the congestion is mainly located on entry links.

6.3.6 Intersection queue length convergence

According to Section 6.3.5, there are several busy intersections that are the main bottlenecks of the network. Among these intersections, three intersections are chosen to explore the effects of route choices on intersection performances. Figure 6.14 shows locations of these three intersections: intersections 5778, 5773, and 5544. All of them are close to centroids where vehicles are generated, which is one reason these intersections are the main bottlenecks. One difference between intersection 5544 and the other two intersections is that all vehicles originated from the centroid (marked with red circles) near intersection 5544 have to pass this intersection. For intersection 5773 and intersection 5778, their closest centroid (marked with the red circle) is located on the link between them. Vehicles originated from this centroid need to pass through either intersection 5778 or intersection 5773.

Figure 6.15 shows the average queue lengths of three intersections in different iterations when the demand is 80000 vehicles. For intersection 5544, the queue length is not significantly affected by the change in path assignment. The traffic volume passing this intersection from its closest centroid does not change, so the variation in the queue length is mainly caused by the redistributed flow from other centroids. For intersection 5773, the queue length increases as the iteration number increases, which indicates that the paths including this intersection are used by more and more vehicles. Intersection 5778 has the opposite trend as the queue length decreases with the iteration number. One possible reason is that vehicles from their closest centroid choose paths either passing intersection 5778 or intersection 5773. If the traffic assignment result converges, the average intersection queue length will also converge to a constant. Figure 6.16 shows the change in average intersection queue length with the demand. The average queue lengths for three intersections increase with the demand. Among three intersections, the queue length of intersection 5544 increases fastest with the demand as all vehicles from an adjacent centroid pass through it.

Figure 6.14: Bottleneck distribution

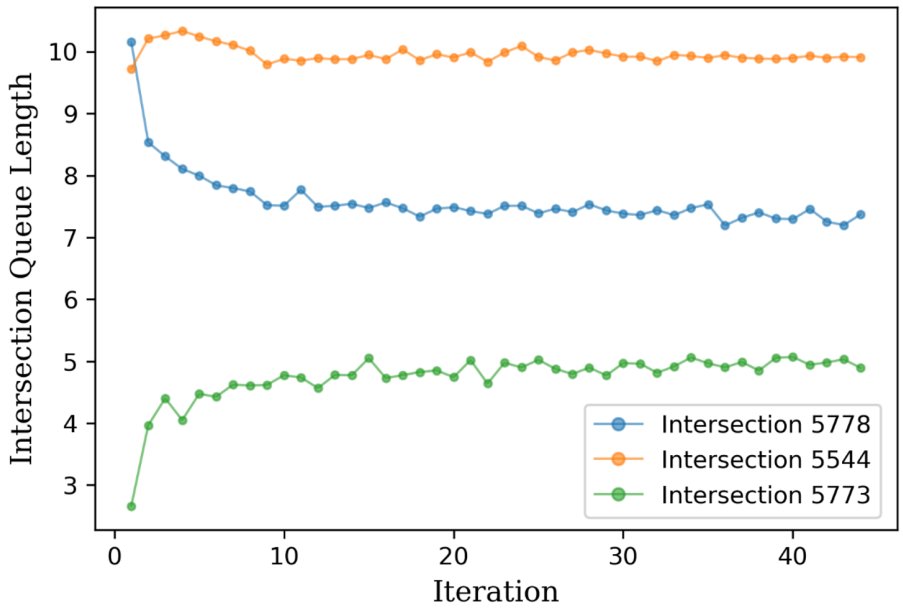


Figure 6.15: Intersection queue length convergence

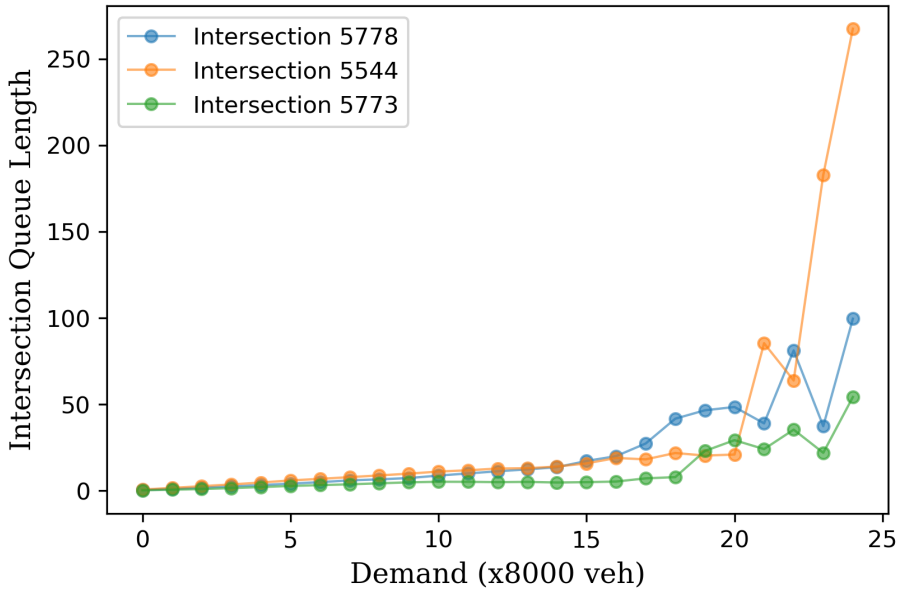


Figure 6.16: Intersection queue length vs demand

6.4 Conclusions

Most studies about max-pressure control test its performance assuming fixed turning proportions without considering the effects of the route choice behavior of road users. It is meaningful to consider road users' route choices and tests their effects on network performances. A simulation-based dynamic traffic assignment model is designed to get the traffic assignment results of the network with max-pressure control. The simulation results show that the network can reach an equilibrium when using max-pressure control when the demand is in the stability region. User equilibrium is not guaranteed when the demand is large and when congestion is generated. The path assignment affects the stability region and further affects the stability of the queue length with the same demand. The average travel time and the total queue length over time between networks using the fixed-time and the max-pressure control are compared, which shows that the max-pressure control has much better performance than the fixed-time control and it can address a larger demand. For each intersection, an increase in demand increases the intersection queue length.

Chapter 7

Microscopic Simulation of Delay-based and Queue-based Max-Pressure Controllers in Realistic Settings

Although the max-pressure control has some good theoretical network-level properties, there are still some problems that may restrict its application in the real world. The first problem is the queue length information for each movement required on the model of Varaiya (2013). This information can be measured when there are exclusive lanes for each movement. If there are combined lanes serving multiple movements, for example, the left-turn and through lane, then the queue length of each movement is not measurable. V2I or V2V technologies are not able to get the next link of a vehicle either. The second problem is that the signal phases used in the real world may include conflicting movements, for example, the left-turn movement and the through movement from the opposite direction. In Varaiya (2013), a phase should only include non-conflicting movements. Both problems can affect the performance of the max-pressure control when it is applied to a real network. A microscopic simulation environment is set up in SUMO and used to test the performance of the max-pressure control. The weight function of the traditional max-pressure control is modified to incorporate measurable link-based or lane-based information. The simulation network has many combined lanes and the existing signal phases for the network, which may activate conflicting movements, are also used.

As previous studies did not evaluate delay-based max-pressure controllers in in a microscopic simulation model, this chapter also includes results from delay-based max-pressure controllers.

7.1 Queue-based and delay-based decentralized controllers

This section describes extensions of the max pressure and proportionally fair controllers used in this chapter. There are seven decentralized controllers used in total: five max-pressure controllers, in which three are queue-based, and two are delay-based, and two proportionally fair controllers, in which one is queue-based and the other on is delay-based.

7.1.1 Max-pressure (MP) controller

The aggregated link information (x_i, τ_i) is used for max-pressure control. As a real network often has combined lanes where more than one movements share one lane, it is hard to know the number of vehicles for each direction.

Max-pressure controller 1: Cyclic queue-based max-pressure controller

This controller uses queue length information. The weight for a traffic movement (i, j) is the difference of queue lengths on upstream and downstream links, as shown in equation (7.2). This weight function is used in Wongpiromsarn et al. (2012), which does not need turning ratios.

$$w_{ij}(t) = x_i(t) - x_j(t) \tag{7.1}$$

The weight of each phase is the sum of weights of all activated movements in this phase multiplying the saturation rate of this phase μ_ϕ.

$$w_{\phi_r}(t) = \mu_{\phi_r} \sum_{(i,j)\in\phi_r} w_{ij}(t) \tag{7.2}$$

The saturation flow rate of a phase is calculated based on saturation rates of lanes used by this phase and the operation of left-turn movements using equations in (Garber and Hoel, 2014). Saturation rates are calculated for types of lanes, including exclusive lanes for through, left-turn, and right-turn movements, shared lanes for left-turn and through movements, shared lanes for through and right movements, and left-turn lanes activated in permissive phases. The total green time G is calculated by using the cycle C subtracted by the total yellow Y and all-red time R, as shown in equation (7.3). In this chapter, cyclic controllers use the same cycle length, yellow time, and all-red time as those in the fixed-time control.

$$G = C - Y - R \tag{7.3}$$

Using the weight of each phase, the green time is assigned proportionally to each phase following the policy in Le et al. (2015), as shown in equation (7.4). η controls the distribution of green time. With a smaller value for η, the green time is more evenly distributed on each phase.

$$G_\phi = G\frac{\exp\{\eta w_\Phi\}}{\sum_{\phi\in\Phi}\exp\{\eta w_\phi\}} \tag{7.4}$$

Le et al. (2015) used a different weight function, as shown in equation (7.5). p_{ij} is the turning ratios from link i to link j. Their intersections used split phasing, which means each phase will activate all movements from a link and different links are served in different phases. Equation (7.5) calculates the weight of a phase activating all movements from a

link so the last term in equation (7.5) is the summation of the weighted queue lengths of all downstream links. In this study, there are phases that activate some movements from a link instead of activating all movements, so we use equations (7.1) and (7.2) to calculate the weight rather than using equation (7.5).

$$w_\phi(t) = \sum_{(i,j)\in\phi} \mu_i \left(x_i(t) - \sum_{j\in\Gamma_i^+} p_{ij} x_j(t) \right) \tag{7.5}$$

Max-pressure controller 2: Non-cyclic queue-based max-pressure controller

When non-cyclic signal control is used, there is no guarantee that each phase will be activated in order. Only the phase with the maximum weight will be activated for a time interval, as shown in equation (7.6). The duration of the time interval varies in studies, such as 5 sec (Sun and Yin, 2018; Wu et al., 2017), 10 sec (Rey and Levin, 2019), and 15 sec (Chen et al., 2020; Gregoire et al., 2014b). In this section, two time step lengths are tested, including 5 and 15 sec. The lengths of the yellow phase and the all-red phase are set to be 3 sec and 2 sec respectively. If two activated phases in two consecutive time intervals are the same, then the yellow phase and the all-red phase between them will be skipped.

$$\phi^\star(t) = \arg\max \{w_\phi(t)\} \tag{7.6}$$

It is hard to tell whether the non-cyclic or cyclic controllers are better. With the non-cyclic controller, some useless phases can be skipped, which is its advantage over the cyclic controller. However, if there are two conflicting movements that both have long queues, then the non-cyclic controller may activate these two movements alternately as the length of the time interval is too short to clean the queue for either movement. Then more yellow phases and all-red phases are activated per time interval, which reduces intersection throughput. Moreover, with non-cyclic controllers, vehicles from a movement with a small demand may have large delays if the demand for its conflicting movement is large. Movements

with a short queue may not be activated for a long time.

Max-pressure controller 3: Cyclic delay-based max pressure controller

For this controller, the total delays of vehicles on the upstream and downstream links related to a turning movement are collected, as shown in equation (7.7), where τ_i is the total vehicle delay on link i.

$$\tau_i(t) = \sum_{v \in \mathcal{V}_i(t)} \tau_v(t) \tag{7.7}$$

$$w_{ij} = \tau_i(t) - \tau_j(t) \tag{7.8}$$

After the weight for each movement is calculated, equations (7.2), (7.3), and (7.4) are used to calculate the green time for each phase.

Max-pressure controller 4: Non-cyclic delay-based max-pressure controller

This controller is similar to max-pressure control 2, but this controller uses the aggregated link delay information to calculate the weight, as shown in equation (7.8). The phase with the maximum weight will be activated. If two consecutive activated phases are the same, then the yellow phase and the all-red phase will be skipped. The lengths of the yellow phase and the all-red phase are set to be 3 sec and 2 sec.

Max-pressure controller 5: Non-cyclic queue-based jam-aware max-pressure controller

Traditional non-cyclic queue-based max-pressure controllers may produce gridlock in networks (Sun and Yin, 2018), so we applied the jam-aware max-pressure traffic controller

proposed by Gregoire et al. (2014b). They defined a convex link pressure function, as shown in equation (7.9), for getting the weight for a movement rather than directly using the link queue length. Any pressure larger than the congestion threshold will be normalized.

$$P_i(x_i) = \min\left(1, \frac{\frac{x_i}{K_\infty} + \left(2 - \frac{K_i}{K_\infty}\right)\left(\frac{x_i}{K_i}\right)^m}{1 + \left(\frac{x_i}{K_i}\right)^{m-1}}\right) \tag{7.9}$$

In equation (7.9), K_∞ and m are parameters that determine the shapes of function $P_i(x_i)$. K_i is the maximum number of vehicles that can stay on link i. A large value of the second term indicates that a large proportion of the link space is occupied by vehicles. Then the weight of a movement is calculated by $w_{ij} = P_i(x_i) - P_j(x_j)$. The phase weight is calculated using equation (7.2). The phase with the largest weight will be activated, which is similar to the non-cyclic queue-based max-pressure controller.

7.1.2 Proportionally-fair (PF) Controller

Proportionally-Fair controller is also a distributed controller originally applied to communication and computer networks (Pióro and Medhi, 2004), but this controller has not been proven to have maximum stability in traffic networks in existing studies. In this thesis, the original form of the PF controller is modified to incorporate traffic data. The implementation of the PF controller needs the queue length or the aggregated delay associated with the lanes activated each phase. The green time allocation under PF controller of phase ϕ is equal to the value of θ_ϕ corresponding to the optimal solution of the following convex

problem:

$$\begin{aligned} \text{maximize} \quad & \sum_{\phi \in \Phi} w_\phi(t) \log \theta_\phi \\ \text{subject to} \quad & \theta_\phi \geq 0, \quad \forall \phi \in \Phi \\ & \sum_{\phi \in \Phi} \theta_\phi = 1 \end{aligned} \tag{7.10}$$

This means that the green time allocated to a phase ϕ under PF policy is proportional to the queue length or the aggregate delay associated with ϕ.

Proportionally-Fair Controller 1: Delay-based proportionally fair controller

The weight of a phase is the aggregated delay on lanes activated in this phase. The aggregated delay on activated lanes are calculated with equation (7.11).

$$\tau_{i_n}(t) = \sum_{v \in \mathcal{V}_{i_n}(t)} \tau_v(t) \tag{7.11}$$

Then the weight of a phase is calculated by equation (7.12).

$$w_\phi(t) = \tau_\phi(t) = \sum_{n \in \mathcal{N}_\phi} \tau_{i_n}(t) \tag{7.12}$$

With phase weights, the proportion of total green time assigned to each phase is calculated by solving the nonlinear programming in (7.10).

Proportionally-Fair Controller 2: Queue-based proportionally fair controller

For this controller, the weight of each phase is the total queue length on activated lanes. The calculation of the green time is similar to the PF controller using aggregated delays. The only difference is that we use the lane queue length $x_{i_n}(t)$ instead of lane aggregated

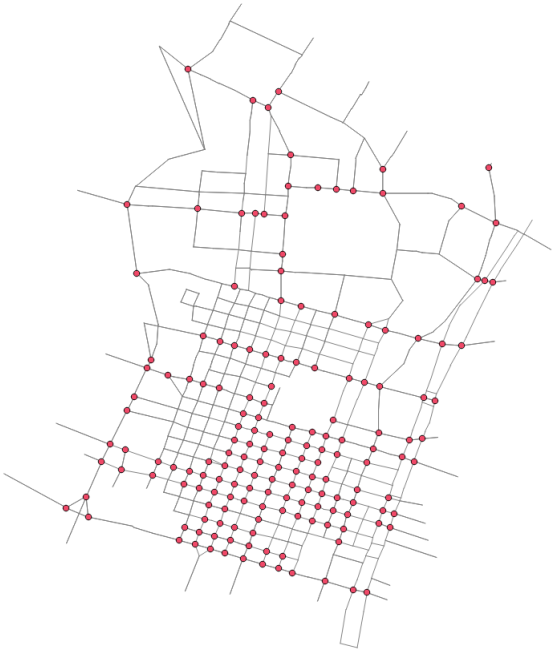


Figure 7.1: Signalized intersections (red dots) in the test network

delay $\tau_{i_n}(t)$.

7.2 Microscopic simulator setup in SUMO

The proposed traffic signal control is implemented in an open-source traffic simulation package called Simulation of Urban Mobility (SUMO) (Lopez et al., 2018), which is a microscopic and continuous traffic simulation package designed to handle large networks.

In this section, a network of Downtown Austin is used for simulation, which includes 158 zones, 506 intersections (with 167 signalized intersections), and 1186 links. Figure 7.1 shows the distribution of signalized intersections in the network. The south part of the network, where the CBD locates, has many more signalized intersections than the north part of the network.

To create the Downtown Austin network in SUMO, information of nodes, links, and

intersection control is needed. As this chapter aims to test the performance of the max-pressure control and the proportionally fair control with a realistic setting, we import the existing signal control for the Downtown Austin model. The max-pressure control and the proportionally fair control can only determine the activation time of each phase. Existing imported network and phase information have some characteristics that may disadvantage the application of the max-pressure control.

1) Nearly 80% of all links include shared lanes serving multiple movements, such as a shared lane for both left and through movements. Because turning movements of vehicles on a combined lane is not known, it is hard to get the actual turning ratios, which are used by the max-pressure controls in many studies to calculate the weight (Varaiya, 2013; Xiao et al., 2014). Only aggregated link information is used for calculating the weight of each movement, so turning ratios are not needed.

2) Max-pressure control or proportionally-fair control prefer using phases that does not include conflicting movements, for example, traditional dual-ring controllers with protected phases for left-turn movements. However, in real world, there are many phases that activate conflicting movements, for example, a phase activating both left-turn movements and through movements in the opposite direction. Figure 7.2 shows a phase that activates movements from both northbound and southbound approaches. The serving of the shared left-turn lane on the northbound approach is blocked by the through movement from the southbound approach, so the long queue on the shared left lane may not be cleared in time.

3) A movement may be activated repeatedly in existing phases. For example, for the intersection shown in Figure 7.2, there may be a phase activating all movements from the northbound approach and a phase activating of all movements from both southbound and northbound approaches. The first phase is better for evacuating traffic on the shared left-turn lane on the northbound approach than the second phase because the through movement from the southbound approach is not activated in the second phase. The existing traffic signal keeps both phases because the northbound approach may have a large traffic volume

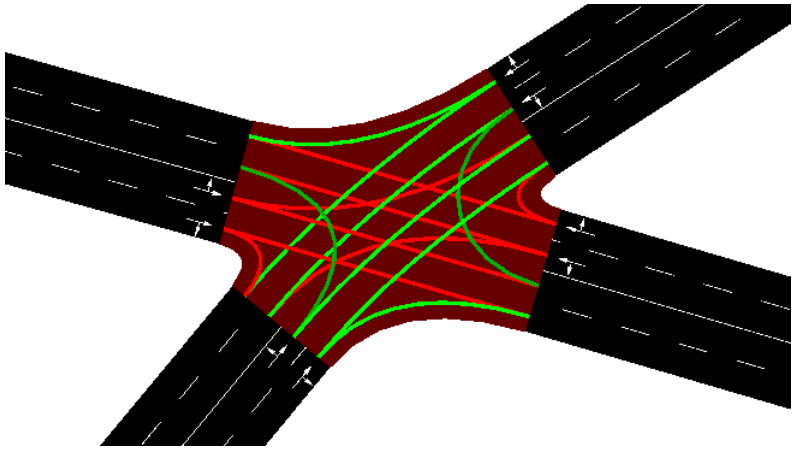


Figure 7.2: A phase related to two incoming links

and needs longer serving time. However, in the max-pressure control or proportionally fair control, whenever there is any vehicle on the southbound approach, the second phase related to two entry approaches will always have a larger weight than the first phase that is related only to the northbound approach.

Besides the network geometry information and the signal control information, the vehicle path information and the demand information should also be imported to the SUMO model. The paths connecting a pair of origin and destination nodes and their probabilities are calculated by the result of dynamic traffic assignment in a mesoscopic simulation. In dynamic traffic assignment, the entire simulation period is divided into multiple assignment intervals. For each pair of the origin and the destination nodes and for each assignment interval, there is a set storing used paths. In every iteration, a new-calculated shortest path will be added to the set if it is not in the set, and the probabilities for all paths in the path set will be updated, which will change the traffic flow distribution on the network and the travel time in the next iteration. The network reaches a user equilibrium if all paths in each path set have the same travel times. The mesoscopic simulation reaches a gap smaller than 1% in the end, which indicates that the network is close to user equilibrium.

For each vehicle in the microscopic simulation, a random number between 0 and 1 is

generated to determine which path in the path set is selected by the vehicle. The path probability does not change with the actual travel time during the microscopic simulation. This setting is consistent with the assumption in the proof of network stability in most studies on the max-pressure control (Le et al., 2015; Varaiya, 2013). In their study, they assume the average turning ratio for each movement is constant over time, which is equivalent to the assumption that the path assignment in the network is constant.

SUMO interface "Traci" is used to implement traffic controllers proposed in previous sections. Traci allows users to set traffic control at every time step. For controllers using the queue length, the number of stopping vehicles on each lane is extracted by Python methods. For controllers using vehicle delays, there is no existing method in Traci that directly measures the cumulative waiting time of a vehicle on a link, so we extract lists of vehicles on entry links of each intersection with Traci, and then store and update the delay information of each vehicle in a separate dictionary in every time step.

7.3 Simulation results

In this section, we test the performance of eight controllers: fixed-time controller (FT), cyclic queue-based max-pressure controller (MPQ-Cyclic), non-cyclic queue-based max-pressure controller (MPQ-NonCyclic), cyclic delay-based max-pressure controller (MPD-Cyclic), non-cyclic delay-based max-pressure controller (MPD-NonCyclic), non-cyclic queue-based jam-aware max-pressure controller (MPQ-JA-NonCyclic), queue-based proportionally fair controller (PFQ), and delay-based proportionally fair controller (PFD). Abbreviations of these seven controllers rather than the full names will be used in plots in this section. Three demand levels, including small, moderate, and high demands, are used in the simulation.

7.3.1 Simulation results with a small demand

In this section, a small demand is used for testing the performances of all controllers. There are 11806 vehicles entering the network in a 2-hour time interval in which the first hour serves as the warm-up session. Figure 7.3 shows variations of the total number of stopped vehicles with the fixed-time controller, queue-based and delay-based controllers. When the demand is small, there is no significant difference between the queue-based and the delay-based controller in the same figure. For example, Figure 7.3a shows that the total number of stopped vehicles for three non-cyclic controllers are similar. Figure 7.3b and Figure 7.3c show the small difference between queue-based and delay-based controllers. In this section, queue-based controllers use aggregated link delays. When the demand is small, all vehicles can pass through the intersection in a short time. When the waiting time for every vehicle is small, there will be a high correlation between the link queue and the aggregated delays for all vehicles on the link.

We can also observe that the performance of non-cyclic controllers is much better than cyclic controllers. The better performance of non-cyclic controllers may largely result from its short activation time interval. The length of the time interval is 15 sec, which is much smaller than all cycle lengths used for cyclic controllers. For cyclic controllers, the input data is collected at the start of each phase. For new-arrived vehicles at the start of a cycle, they may need to wait for a cycle before their movements are activated. For example, consider an intersection with a cycle of 90 sec with two conflicting movements (m_1, and m_2). At time 0, $x_{m_1} = 10$ and $x_{m_2} = 2$. The queue or delay information is measured at 0 sec. Suppose the calculated phase time for m_1 is 80 sec, and the phase time for m_2 is 2 sec with the remaining time are for yellow and all-red phases. The cycle starts with the phase activating m_2. Suppose a vehicle arrives at the 6th sec and joins m_2. This vehicle needs to wait for 84 sec if m_2 is activated in the next time step. In this condition, if a non-cyclic controller is used, this vehicle only needs to wait for 9 sec if the next phase

Table 7.1: Average travel times with a demand of 5900 trips per hour

Controller	Fixed-time	MPQ-Cyclic	MPQ-NonCyclic	MPD-Cyclic	MPD-NonCyclic	PFQ	PFD	MPQ-JA-NonCyclic
Average travel time (sec)	348.7	334.5	261.2	345.4	261.9	399.0	410.1	277.2

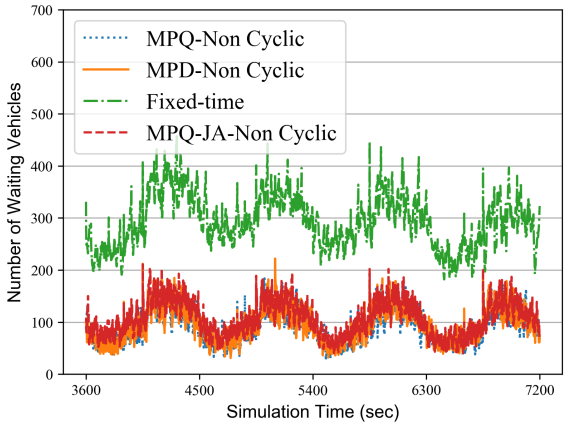
is for m_2. However, when there are two conflicting movements, both with long queue lengths, non-cyclic controllers may switch between two phases to activate two movements frequently, which causes a waste of green time. Similarly, for cyclic controllers, if the calculated duration of a green phase is way longer than the duration needed to evacuate the queue, it also causes a waste of green time. Moreover, cyclic max-pressure controllers have better performance than proportionally fair controllers and the fixed-time controller. The two proportionally fair controllers have similar performances as the fixed-time controller.

Figure 7.4 and Table 7.1 show the average travel times of queue-based and delay-based max-pressure and proportionally fair controllers. The average travel time is calculated based on travel times of vehicles departing at different times. We observe that the average travel time of the fixed-time controller for vehicles departing between the 60th minutes and the 75th minutes is large, which indicates that vehicles departing at this time meet traffic congestion at some intersections along their paths. In Figure 7.4a, the average travel times of three non-cyclic max-pressure controllers are smaller than that of the fixed-time controller and all cyclic controllers. In Figure 7.4b, the average travel times of two cyclic max-pressure controllers are close to the average travel time of the fixed-time controller. Two proportionally fair controllers have large travel times than the fixed-time controller after 75 minutes.

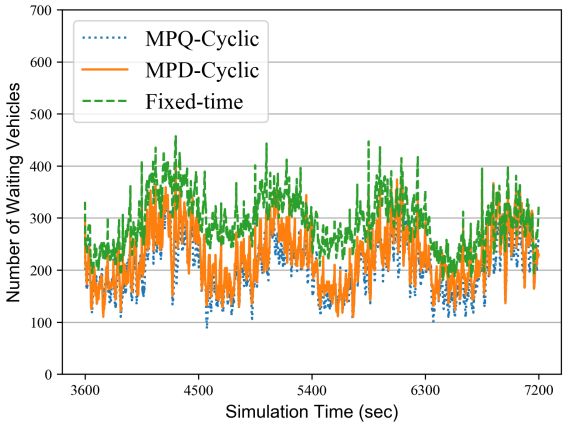
7.3.2 Simulation results with a moderate demand

In this section, a moderate demand is used to test the performance controllers. There are 26400 vehicles entering the network in two hours.

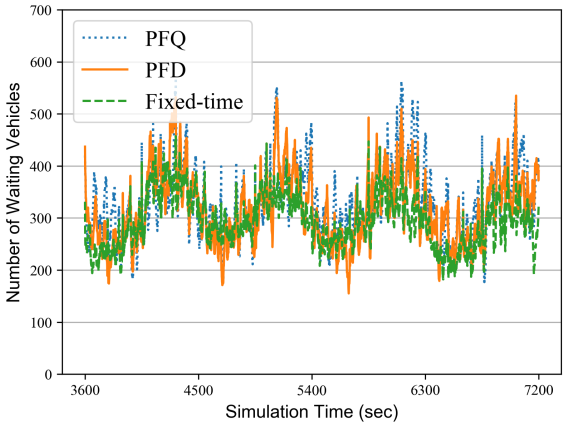
Figure 7.5 show the variation of the total number of stopped vehicles in the network.



(a) MPQ-Non Cyclic vs MPD-Non Cyclic vs MPQ-CA-Non Cyclic

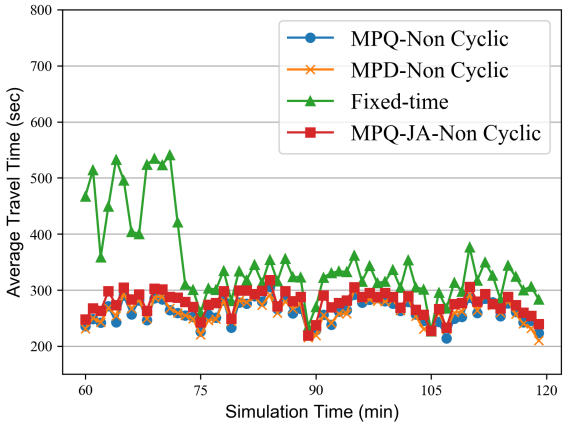


(b) MPQ-Cyclic vs MPD-Cyclic

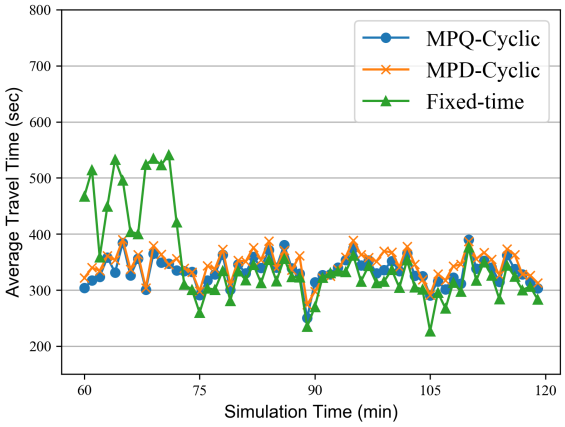


(c) PFQ vs PFD

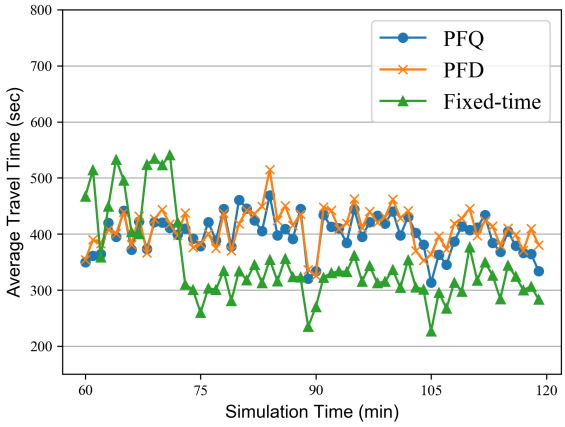
Figure 7.3: Total number of stopped vehicles with a demand of 5900 trips per hour



(a) MPQ-Non Cyclic vs MPD-Non Cyclic vs MPQ-CA-Non Cyclic



(b) MPQ-Cyclic vs MPD-Cyclic

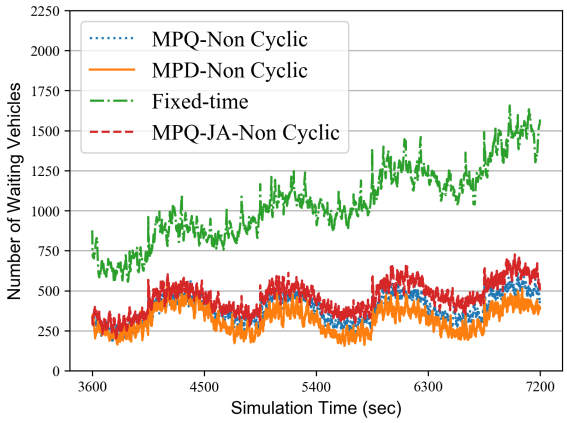


(c) PFQ vs PFD

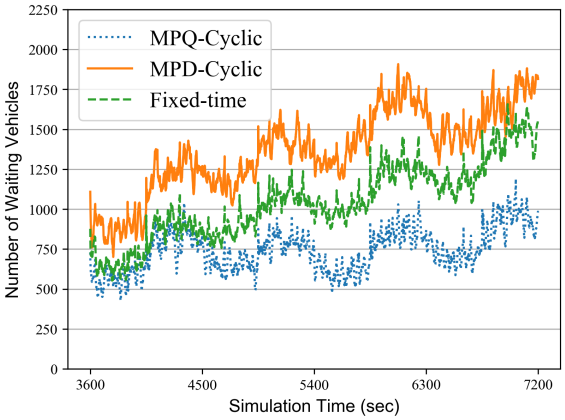
Figure 7.4: Average travel times with a demand of 5900 trips per hour

The delay-based cyclic max-pressure controller, the queue-based proportionally fair controller, and the fixed-time controller cannot stabilize the total number of stopped vehicles under this demand level. For these controllers, the value rises on average. The delay-based cyclic max-pressure controller has a larger number of stopped vehicles than the fixed-time controller. The delay-based cyclic max-pressure controller prioritizes a group of vehicles with a larger aggregated delay. Suppose there are two conflicting movements m_1 and m_2. m_1 has 10 vehicles and each of them has a delay of 2 seconds. m_2 has 2 vehicles and each of them has a delay of 12 seconds. Then the movement with fewer vehicles will be assigned a longer green time interval, which may affect the throughput of the network. The non-cyclic delay-based controller can stabilize the network queue length is because it uses a short time interval and the controller may activate two conflicting movements alternately with a high frequency. All the other controllers can stabilize the total number of stopped vehicles. For these controllers, the variation is periodic, and the value tends to fluctuate between two numbers.

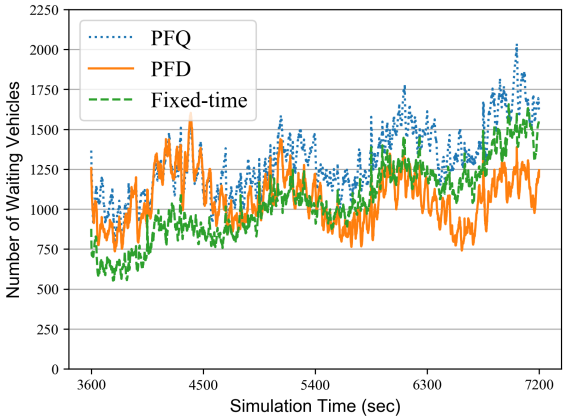
The average travel times for arrived vehicles departing at every minute are extracted. Figure 7.6 and Table 7.2 show the average travel times in simulation with different controllers. For three non-cyclic controllers shown in Figure 7.6, there is no significant difference between their travel times, and their travel times are much shorter than the travel time of the fixed-time controller. The queue-based non-cyclic jam-aware max-pressure controller has a slightly larger travel time than the other two non-cyclic controllers. For two cyclic max-pressure controllers, the delay-based one has a higher travel time than the fixed-time controller, while the queue-based one has a much shorter travel time than the fixed-time controller. For two proportionally fair controllers, the delay-based one has a larger travel time than the queue-based one. Moreover, all non-cyclic controllers have much shorter travel times than cyclic controllers.



(a) MPQ-Non Cyclic vs MPD-Non Cyclic vs MPQ-CA-Non Cyclic



(b) MPQ-Cyclic vs MPD-Cyclic

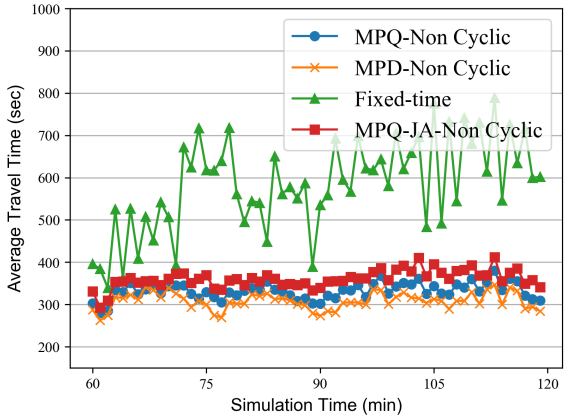


(c) PFQ vs PFD

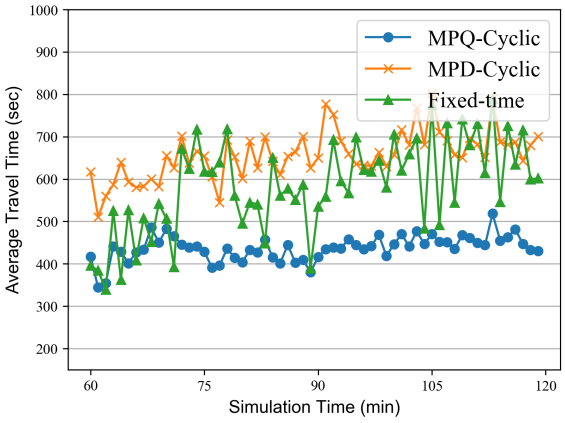
Figure 7.5: Total number of stopped vehicles with a demand of 13200 trips per hour

Table 7.2: Average travel times with a demand of 13200 trips per hour

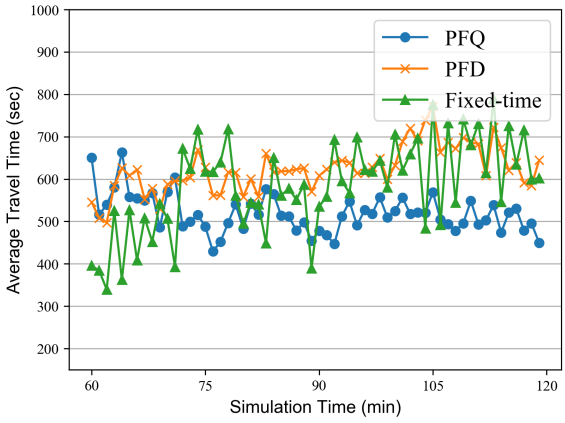
Controller	Fixed-time	MPQ-Cyclic	MPQ-NonCyclic	MPD-Cyclic	MPD-NonCyclic	PFQ	PFD	MPQ-JA-NonCyclic
Average travel time (sec)	584.3	437.2	331.8	657.3	308.3	519.2	623.5	360.9



(a) MPQ-Non Cyclic vs MPD-Non Cyclic vs MPQ-CA-Non Cyclic



(b) MPQ-Cyclic vs MPD-Cyclic



(c) PFQ vs PFD

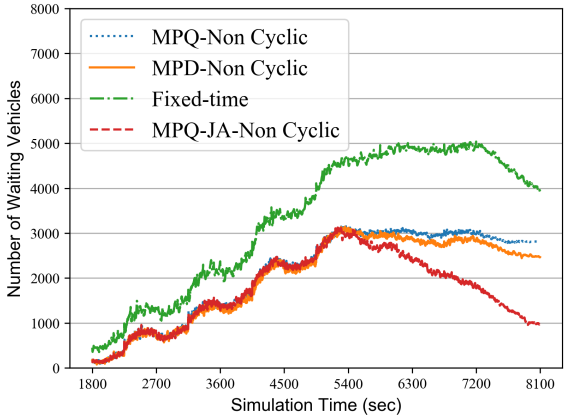
Figure 7.6: Average travel times with a demand of 13200 trips per hour

7.3.3 Simulation results with a large demand

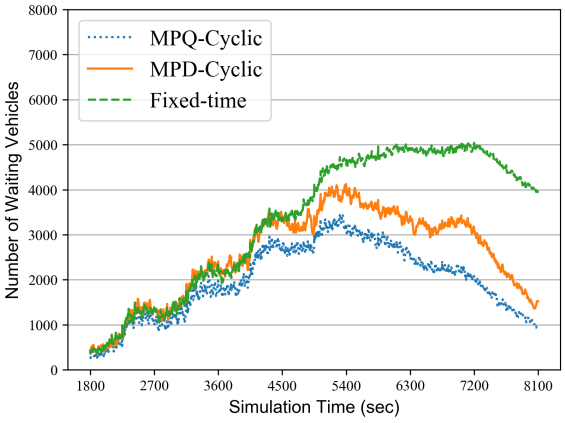
In existing studies, properties of the max-pressure control and the proportionally fair control are derived under the prerequisite that the demand is in the stability region. It is unknown how the performance of these controllers will be when the demand is outside of the stability region. In this section, controller performances are tested with a large demand with which the network is unstable. The first half-hour is the warm-up session, and there are 4870 vehicles entering the network. The warm-up session is followed by a one-hour period with a high demand of 20950 vehicles. This one-hour period is followed by another one-hour period with a small demand in order to mitigate the congestion with 5600 vehicles entering the network.

Figure 7.7 shows the variations of the total number of stopped vehicles for all controllers. The fixed-time controller has the worst performance, with its maximum queue length reaches 5000. Three non-cyclic controllers reach their largest number of stopped vehicles about 3000 vehicles at the 5400 sec. They have smaller queues than all of the cyclic controllers before the numbers of other controllers drop to 3000. The total numbers of the stopped vehicles of the queue-based and the delay-based non-cyclic max-pressure controllers stay constant at 3000 for a long time because gridlock is generated in the network with these two controllers. However, the queue-based jam-aware non-cyclic controller does not produce a gridlock even with high demand, and its total number of stopped vehicles drops gradually after the 5400 sec. Neither of the two cyclic controllers produces a gridlock because all phases are activated by cyclic controllers so that vehicles from all directions are able to move to their downstream links. Among all controllers, the non-cyclic queue-based jam-aware max-pressure controller, the cyclic queue-based max-pressure controller, and the delay-based proportionally fair controller have the best performance.

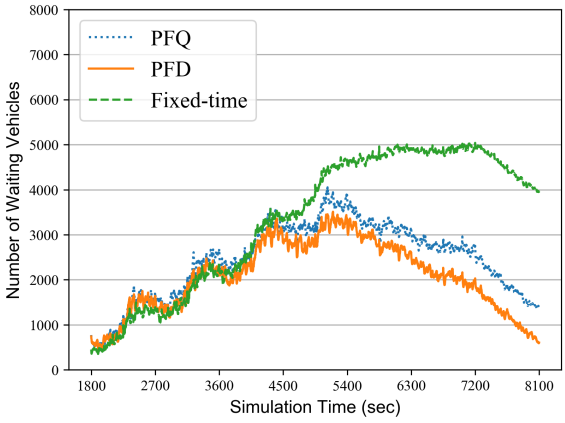
Figure 7.8 and Table 7.3 show the average travel times of controllers. Overall, the average travel time for the fixed-time controller is the largest. For both pairs of cyclic controllers



(a) MPQ-Non Cyclic vs MPD-Non Cyclic vs MPQ-CA-Non Cyclic

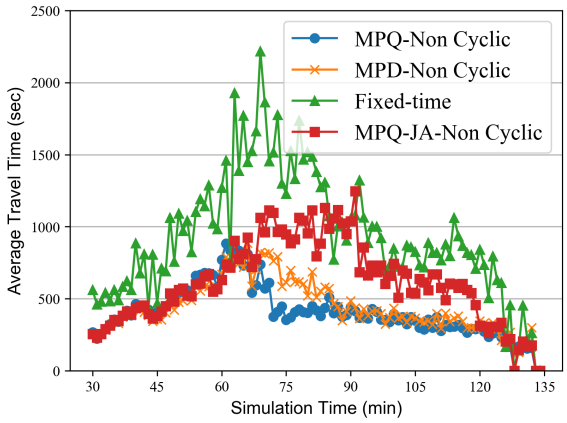


(b) MPQ-Cyclic vs MPD-Cyclic

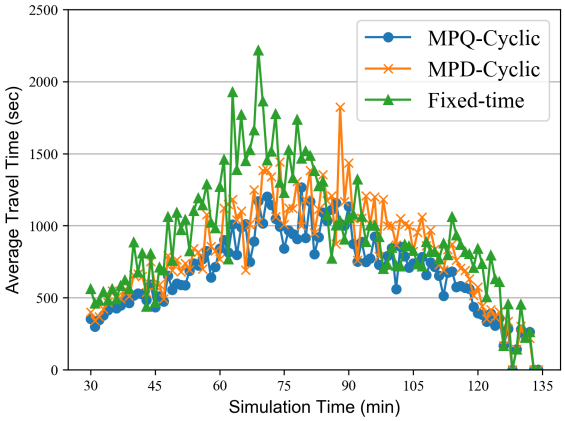


(c) PFQ vs PFD

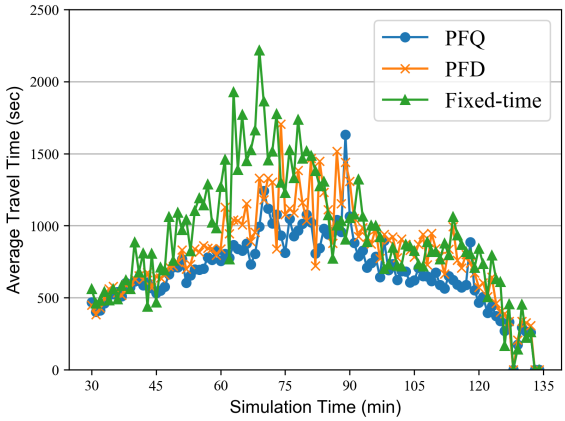
Figure 7.7: Total number of stopped vehicles with a demand of 20950 trips per hour



(a) MPQ-Non Cyclic vs MPD-Non Cyclic vs MPQ-CA-Non Cyclic



(b) MPQ-Cyclic vs MPD-Cyclic



(c) PFQ vs PFD

Figure 7.8: Average travel times with a demand of 20950 trips per hour

show similar performances and queue-based cyclic controllers have slightly better performances than delay-based cyclic controllers under a large demand. The average travel times for all arrived vehicles with the queue-based and the delay-based non-cyclic max-pressure controllers are much smaller than the queue-based non-cyclic jam-aware controller after the network becomes congested, but the travel times of the former two controllers are only calculated for arrived vehicles. Figure 7.9 shows the throughput of all controllers when the network is congested. The maximum observed throughput of the fixed-time control

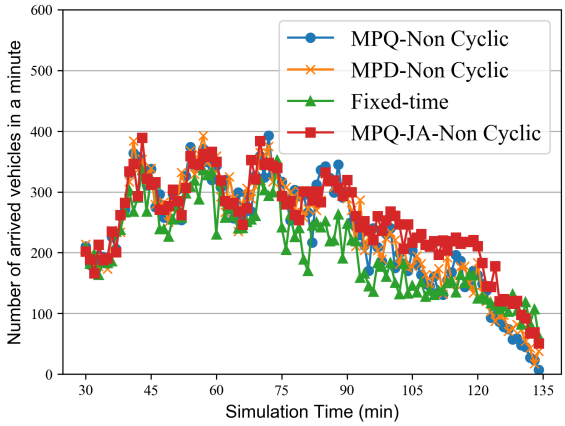
Table 7.3: Average travel times with a demand of 20950 trips per hour

Controller	Fixed-time	MPQ-Cyclic	MPQ-NonCyclic	MPD-Cyclic	MPD-NonCyclic	PFQ	PFD	MPQ-JA-NonCyclic
Average travel time (sec)	1144.4	857.1	440.3	1038.0	508.3	826.3	1009.3	796.8

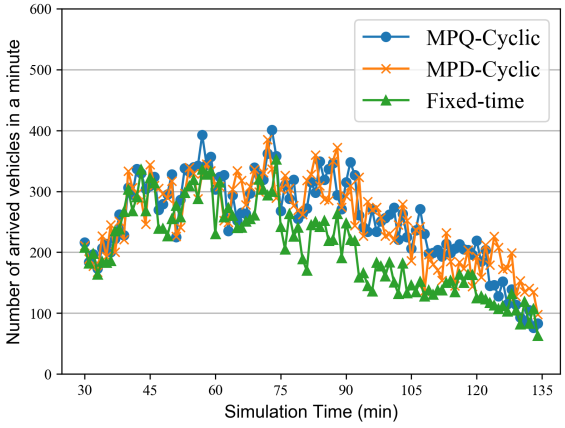
is 350 vehicles per minute. For three non-cyclic controllers, their maximum throughput values can reach 400 vehicles per minute, which is the same as two cyclic queue-based controllers. Two cyclic delay-based controllers have smaller values for their maximum throughput, which are between 370–380 vehicles per minute. From Figure 7.9a, we can observe that the throughput of the non-cyclic queue-based and the delay-based max-pressure controllers drops to 0, indicating there is gridlock.

Figure 7.10b shows the location of gridlock. Link 18394 is a one-way road going west. As shown in Figure 7.10b, all vehicles on link 18394 are turning left to link 5153, which is already full. Vehicles on link 105153 cannot move forward to the downstream link 105307, because the controller is activating the phase for link 18394 which has a larger weight. However, as the downstream link 5153 for vehicles on link 18394 is full, vehicles on 18394 cannot move. The queue on link 5153 is propagated from link 5266, on which vehicles cannot move forward because the controller is activating the phase for link 18162. Most vehicles on link 18162 are turning right to link 105266. As link 105266 is filled by the queue propagated from link 105153, vehicles on link 18162 cannot move, which increases the queue on this link. Then the queues on link 18162 and 18394 will be longer and longer.

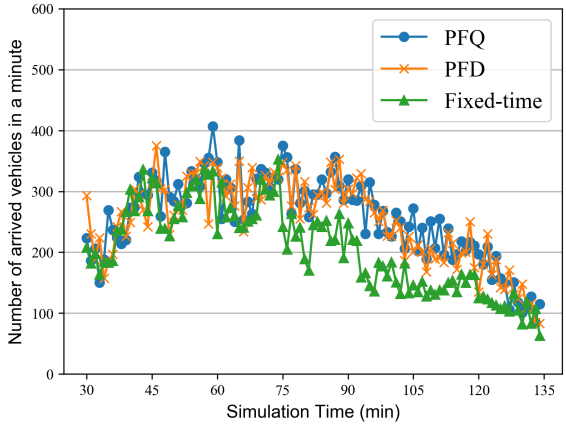
Figure 7.10a shows the same location controlled by the jam-aware max-pressure controller. For this controller, the link's pressure relates to the proportion of the occupied space on the link. When the queues on both link 105153 and link 5307 fill the link, vehicles on link 105153 can move to link 105307 because the phase for the movement from link 105153 to link 105307 and the movement from link 5307 to link 5153 has a larger weight than the phase for the movement on link 18394.



(a) MPQ-Non Cyclic vs MPD-Non Cyclic vs MPQ-CA-Non Cyclic

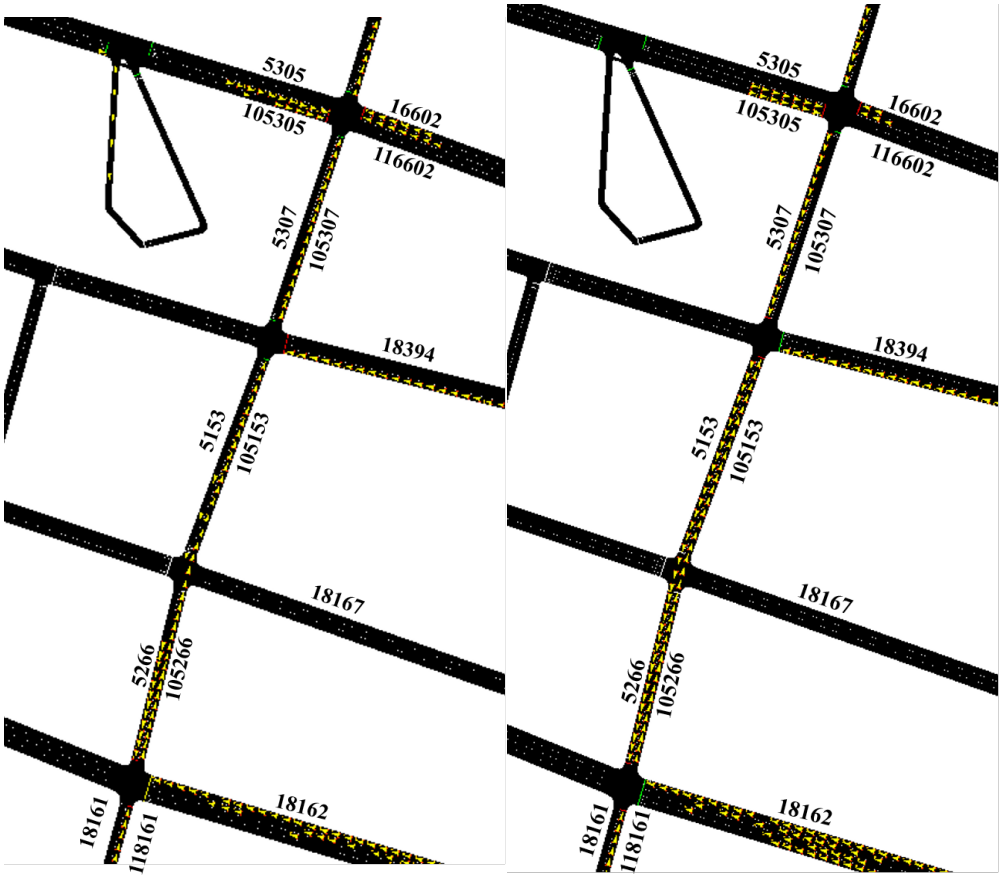


(b) MPQ-Cyclic vs MPD-Cyclic



(c) PFQ vs PFD

Figure 7.9: Throughput with a demand of 20950 trips per hour



(a) no grid-lock with the capacity-aware controller

(b) grid-lock with the non capacity aware non-cyclic controller

Figure 7.10: Gridlock in the network with non-cyclic controllers

Table 7.4: Average number of stopped vehicles of non-cyclic controllers

	MPQNC-15sec	MPQNC-5sec	MPQNCaware-15sec	MPQNCaware-5sec
small demand	91.4	60.9	105.1	74.5
moderate demand	384.7	694.0	454.8	800.9
large demand	2119.9	2698.5	1725.4	2509.0

Table 7.5: Average travel times of non-cyclic controllers

	MPQNC-15sec	MPQNC-5sec	MPQNCaware-15sec	MPQNCaware-5sec
small demand	261.2	243.5	277.2	261.3
moderate demand	331.8	464.0	360.9	514.8
large demand	265.3	345.3	288.0	382.8

7.3.4 The effect of the time step size for non-cyclic controllers

Time step size is an important parameter for non-cyclic controllers. To test the effect of time step size on the performance of non-cyclic controllers, we run simulations using the non-cyclic queue-based max-pressure controller and the non-cyclic queue-based jam-aware max-pressure controller with a time step size of 5 sec. The simulation results are compared with the results using a time step size of 15 sec.

Table 7.4 and Table 7.5 show the total number of stopped vehicles and average travel times of four controllers. The names ”MPQNC-15sec” and ”MPQNCaware-15sec” represent two non-cyclic queue-based max-pressure controllers used in previous sections. The names ”MPQNC-5sec” and ”MPQNCaware-5sec” represent two non-cyclic queue-based max-pressure controllers with a step size of 5 sec. According to the first row of two tables, Non-cyclic controllers with a smaller time step size have smaller numbers of stopped vehicles and smaller average travel times when the demand is small. If the queue length at an intersection is short and the queue can be served by a short time interval, when the queue is served earlier, the delay is smaller. When the demand gets larger, the non-cyclic controllers with a step size of 5 sec have much larger numbers of stopped vehicles and average travel times than the controllers with a step size of 15 sec. When queues cannot be served

in a short time interval, switching between two phases with a high frequency will reduce intersection throughput because of the time loss between two phases.

7.4 Conclusions

This chapter applies the fixed-time controller, five max-pressure controllers, and two proportionally fair controllers to signalized intersections in a large test network in microscopic simulation. Five max-pressure controllers include the non-cyclic queue-based max-pressure controller, the delay-based max-pressure controller, the cyclic queue-based max-pressure controller, the cyclic delay-based max-pressure controller, and a non-cyclic queue-based jam-aware max-pressure controller. Two proportionally fair controllers include one using queues and one using delays.

The test network is built in the microscopic simulation package SUMO with node and link information in Downtown Austin. Existing signal control for 167 signalized intersections is imported to the model. SUMO interface "Traci" is used to extract detector data and apply signal controllers during the simulation. Three demand levels, including a small demand, a moderate demand, and a large demand, are tested. The total number of stopped vehicles and the average travel time are extracted after the simulation.

When the demand is small, the three non-cyclic controllers have the best performance. Two cyclic max-pressure controllers have slightly better performances than the fixed-time controller, while two proportionally fair controllers have similar performances to the fixed-time controller. When a moderate demand is used, two cyclic delay-based controllers have worse performance than the fixed-time controller, and three non-cyclic controllers have the best performance. When the demand is large, the only non-cyclic controller that does not produce gridlock is the non-cyclic queue-based jam-aware max-pressure controller, although this controller has a larger average travel time than the other two non-cyclic controllers. Overall, for all three demand levels, there are two controllers that have acceptable

travel times and a total number of stopped vehicles and do not produce gridlock: the non-cyclic queue-based jam-aware max-pressure controller, and the cyclic queue-based max-pressure controller.

This chapter uses aggregated link information to calculate movement weights without using turning ratio information. Moreover, the controllers used in this chapter does not consider the vehicle distribution on different lanes of a link. Using turning ratio information or lane-based information in the weight calculation may significantly improve the performance of decentralized controllers. In future work, we will propose a max-pressure controller using lane-based information and the turning ratio information, which can be applied to a network with shared lanes serving multiple movements.

Chapter 8

Routing Guidance with Max-Pressure Control

Road users' route choice and intersection control are two factors that affect the efficiency of the traffic network that also interact. The signal timing is updated based on traffic volumes, and traffic volumes can be affected by path travel times, including waiting times at intersections and running times on links. It is difficult for road users and traffic engineers to reach a consensus automatically as they follow different objectives. For traffic engineers, they want to develop intersection control algorithms and improve the traffic efficiency of the entire network. For road users, they seek the shortest path to their destinations according to their perceived travel cost information. For distributed traffic signal control policies, when selfish route choices are allowed in the model, signal control can not fully make use of the available capacity of the network (Smith, 2015). User equilibrium (UE) can be achieved in networks with selfish route choices, which is a realistic state but not an ideal state of the network. In the Braess paradox, adding an additional link to a congested network may increase the travel time for all paths if UE is assumed (Braess, 1968). If route choices of road users can be controlled, then the effectiveness of intersection control can be enhanced.

The development of autonomous driving technologies provides a potential solution to traffic congestion. As autonomous vehicles have smaller reaction times and are equipped with inter-vehicle communication devices, they are expected to increase the road capac-

ity (Levin and Boyles, 2016; Van den Berg and Verhoef, 2016), and they are much easier to be controlled when compared to human-driven vehicles. If vehicles are under control, it is possible to control the route choice of road users and improve the efficiency of the network, for example, reducing the total system travel time. The problem of controlling network flow and routing vehicles is often formulated as a centralized problem (Zhang and Nie, 2018). Centralized problems require network-level information and may face scalability problems for large networks. Therefore, one motivation of the study is proposing a distributed routing guidance algorithm based on the max-pressure control to modify vehicles' routing behaviors.

8.1 Network model

The traffic flow model used in this chapter is similar to that defined in Chapter 3. As destination information is used in the routing guidance algorithm, destination index is added to variables in the traffic flow model defined in Chapter 3. Let $\mathcal{O}$ be a set of destination links, and it is a subset of the set of exiting links $\mathcal{L}_{\text{exit}}$ if some exiting links are not used. Let $x_i^o(t)$ be the number of vehicles heading for destination o on link i. $x_{ij}^o(t)$ is the number of vehicles heading for destination o in the queue of movement (i,j). $y_{ij}^o(t)$ is the flow of vehicles with destination o leaving the queue of movement (i,j). We have $\sum_{o\in\mathcal{O}} y_{ij}^o(t) = y_{ij}(t)$.

$p_{ij}(t)$ is the percentage of vehicles on link i going to link j at time t. $p_{ij}^o(t)$ is the percentage of vehicles heading for destination o on link i going to link j at time t. Let $\bar{p}_{ij}$ be the average turning proportion of vehicles from link i to link j and $\bar{p}_{ij}^o$ is the average turning proportion for vehicles heading for destination o. Let $\bar{\mathbf{p}} = \{\bar{p}_{ij}^o, \forall i \in \mathcal{L}\}$ be a vector of all average turning proportions for the network. P denotes the turning proportion of all movements in the network and $\mathcal{P}$ is a set of all values of P.

Let Φ be the set of all phases. The activation of phase ϕ at time t is denoted by $s_\phi(t) \in \{0,1\}$. The activation of movement (i,j) is represented by $s_{ij}(t) \in \{0,1\}$. $\delta_\phi^{ij} \in \{0,1\}$ is

used to indicate if movement (i, j) is activated in phase ϕ. If $\delta_\phi^{ij} = 1$, then movement (i, j) is activated by phase ϕ. We have $s_{ij}(t) = \sum_{\phi \in \Phi} s_\phi(t)\delta_\phi^{ij}$. $s_n(t)$ denotes the signal control at intersection n at time t and $s(t)$ includes intersection control for all intersections at time t. $s_n = \{s_n(t), t = \{1, 2, ..., T\}\}$ is the intersection control sequence for intersection n and s is the network control. $\mathcal{S}$ is a set of all values of s. Given a intersection control sequence, the average proportion of time that movement (i, j) is activated can be calculated with equation (8.1).

$$\bar{s}_{ij} = \frac{1}{T}\sum_{t=1}^{T}\sum_{\phi \in \Phi} s_\phi(t)\delta_\phi^{ij} = \frac{1}{T}\sum_{t=1}^{T} s_{ij}(t) \tag{8.1}$$

$\bar{y}_{ij}^o$ is the average flow of vehicles with destination o leaving queue (i, j). Let $\bar{y}_{ij} = \sum_o \bar{y}_{ij}^o$ be the average flow leaving the queue of movement (i, j) and $\bar{y}_i = \sum_{j \in \Gamma_i^+} \bar{y}_{ij}$ be the average flow leaving link i. The relation between the average flow and the average turning proportion is shown in equation (8.2). Let $\bar{\mathbf{y}}$ be the vector of $\bar{y}_{ij}^o$.

$$\bar{p}_{ij} = \frac{\sum_{t=1}^{T} y_{ij}(t)}{\sum_{t=1}^{T} y_i(t)} = \frac{\sum_{t=1}^{T} y_i(t)p_{ij}(t)}{\sum_{t=1}^{T} y_i(t)} \tag{8.2}$$

The capacity, which is the maximum service rate of movement (i, j) in phase ϕ, is denoted by Q_ϕ^{ij}. The actual capacity of movement (i, j) is $Q_\phi^{ij} s_\phi$. If movement (i, j) is not activated in phase ϕ, then $Q_\phi^{ij} = 0$. $\hat{Q}_{ij}$ is the maximum service rate of movement (i, j) in all phases.

8.1.1 Intersection control

The weight of a movement (i, j) is defined as the queue length of movement (i, j) subtracted by the weighted average downstream queue length, as shown in equation (8.3).

$$w_{ij}(t) = x_{ij}(t) - \sum_{k \in \Gamma_j^+} p_{jk}(t) x_{jk}(t) \tag{8.3}$$

The optimal control can be calculated with the following nonlinear program:

$$\begin{aligned}
\text{max} \quad & \sum_{\phi \in \Phi} w_\phi(t) s_\phi(t) && \text{(8.4a)} \\
\text{s.t.} \quad & s_\phi(t) \in \{0, 1\}, \quad \forall \phi \in \Phi && \text{(8.4b)} \\
& \sum_{\phi \in \Phi} s_\phi(t) = 1 && \text{(8.4c)} \\
& w_{ij}(t) = x_{ij}(t) - \sum_{k \in \Gamma_j^+} p_{jk}(t) x_{jk}(t) && \text{(8.4d)} \\
& w_\phi(t) = \sum_{(i,j) \in \phi} Q_\phi^{ij} w_{ij}(t) && \text{(8.4e)} \\
& p_{jk}(t) = \frac{\sum_o p_{jk}^o(t) x_j^o(t)}{x_j(t)} && \text{(8.4f)} \\
& p_{jo}^o(t) = 1, \forall o \in \mathcal{O} && \text{(8.4g)} \\
& p_{jn}^o(t) = 0, \forall o, n \in \mathcal{O} && \text{(8.4h)} \\
& p_{jk}^o(t) \geq 0 && \text{(8.4i)} \\
& p_{jk}(t) \geq 0 && \text{(8.4j)} \\
& \sum_{k \in \Gamma_j^+} p_{jk}^o(t) = 1 && \text{(8.4k)} \\
& \sum_{k \in \Gamma_j^+} p_{jk}(t) = 1 && \text{(8.4l)}
\end{aligned}$$

The optimal control solved in equations (8.4) is denoted as $C^\star = \{S^\star, P^\star\}$. In equation (8.4), the objective function is maximizing the total pressure of the intersection, where w_ϕ is the phase weight and s_ϕ is a binary variable indicating the activation of a phase. Only one phase is allowed to be activated at a time period. Constraint (8.4d) represents the weight function for a movement. Constraint (8.4e) represents the weight function for a phase. The phase weight is calculated by adding up the movement weight and capacity of activated movements. Constraints (8.4j) and (8.4i) guarantee the variable for turning proportion is non-negative. Constraint (8.4f) represents the relation between the turning proportion variable without the destination index and the one with the destination index. Constraint (8.4g) force a turning proportion variable with the destination index to be 1 when the downstream link j connects to the destination link o. If downstream link j is connected to an exit link, but this link is not a destination link, then the turning proportion should be set to 0, as shown in constraint (8.4h). Constraint (8.4k) shows that for each destination, the turning proportion variables should sum up to 1. Constraint (8.4l) shows the summation of all turning proportion for movements from the same upstream link should be 1. Program (8.4) is a mixed integer nonlinear program because the objective function includes terms which is the product of a binary variable s_{ϕ_r} and a continuous variable p_{jk}. One simple solution algorithm is enumerating all phases, calculating the value of objective function when this phase is chosen, and then pick the phase with the largest optimized objective function. This algorithm needs m runs to finish the calculation where m is the number of phases. Another way is to linearize the objective function. A new variable z^r_{jk} can be introduced to replaced the term representing the produce of s_{ϕ_r} and p_{jk}. At meanwhile, four additional constraints need to be added for every introduced variable Z^r_{jk}. The parameter $\hat{p}_{jk}$ in constraints (8.5a) and (8.5a) represents the maximum value of p_{jk}, which equals 1 in this program.

$$z^r_{jk} \leq \hat{p}_{jk} s_{\phi_r} \tag{8.5a}$$

$$z^r_{jk} \leq p_{jk} \tag{8.5b}$$

$$z^r_{jk} \geq p_{jk} - (1 - s_{\phi_r})\hat{p}_{jk} \tag{8.5c}$$

$$z \geq 0 \tag{8.5d}$$

8.1.2 Stability

The control policy can be denoted as $\mathcal{C} = \{P, S\}$, where $P \in \mathcal{P}$ and $S \in \mathcal{S}$. Consider a demand vector $\bar{\mathbf{d}} = \{\bar{d}_{ij}, i \in \mathcal{L}_{\text{entry}}\}$, if there exists a control policy $\mathcal{C} = \{P, S\}$ and a flow vector $\bar{\mathbf{y}}$ that satisfy the constrains in equation (8.6), this demand vector will be included by set D. The interior of set D is denoted by D^o.

$$\bar{d}^o_{ij} = \bar{y}^o_{ij} \qquad \forall\, i \in \mathcal{L}_{\text{entry}}, \forall\, o \in \mathcal{O} \tag{8.6a}$$

$$\sum_{h \in \Gamma^-_i} \bar{y}^o_{hi} \bar{p}^o_{ij} = \bar{y}^o_{ij} \qquad \forall\, i \in \mathcal{L}_{\text{int}}, \forall\, o \in \mathcal{O} \tag{8.6b}$$

$$\sum_o \bar{y}^o_{ij} \leq \sum_\phi \bar{s}_\phi Q^{ij}_\phi \qquad i \in \mathcal{L} \tag{8.6c}$$

$$\sum_{i \in \mathcal{L}_{\textbf{entry}}} \sum_{j \in \Gamma^+_i} \bar{d}^o_{ij} = \sum_{k \in \Gamma^-_o} \bar{y}^o_{ko} \qquad \forall o \in \mathcal{O} \tag{8.6d}$$

Proposition 3. *If $\bar{\mathbf{d}} \notin D^o$, the network is unstable under any signal control.*

Proof. If a demand $\bar{\mathbf{d}}$ is not in the stability region, we can still find $\bar{\mathbf{y}}$ and $\bar{\mathbf{p}}$ that satisfy equations (8.6a), (8.6b), and (8.6d), but we can not find any control to stabilize the demand, so inequality (8.6c) is not satisfied for some movements. Let $\mathcal{M}_u$ be the set of these movements, and $|\mathcal{M}_u| \geq 1$. Then there exists $\epsilon > 0$ and we have:

$$\sum_o \bar{y}^o_{ij} \geq \sum_\phi \bar{s}_\phi Q^{ij}_\phi + \epsilon,\ \forall\, (i,j) \in \mathcal{M}_u \tag{8.7}$$

Then the total queue length of the network will accumulate as time goes:

$$\lim_{t\to\infty}\sum_{(i,j)\in\mathcal{M}}\frac{x_{ij}(t)}{t}=\sum_{(i,j)\in\mathcal{M}}\left(\sum_{o}\bar{y}_{ij}^{o}-\sum_{\phi}\bar{s}_{\phi}Q_{\phi}^{ij}\right)\geq\epsilon \tag{8.8}$$

Equation (8.8) indicates that the total queue length still has the trend to increase with time. As the long term average increasing rate of the total queue length is greater or equal than ϵ, then we can find a time moment $t=t^{'}$, when the total queue length equals ϵ. If $t>t^{'}$, we have $\sum_{(i,j)\in\mathcal{M}}x_{ij}(t)>\epsilon t^{'}$. Then we have:

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\sum_{(i,j)\in\mathcal{M}}x_{ij}(t)>\lim_{T\to\infty}\frac{1}{T}\sum_{t=t^{'}}^{T}\sum_{(i,j)\in\mathcal{M}}x_{ij}(t)>\lim_{T\to\infty}\frac{1}{T}\sum_{t=t^{'}}^{T}\epsilon t=\infty \tag{8.9}$$

Equation (8.9) shows that the time average queue length is not bounded. □

Proposition 4. *If $\bar{\mathbf{d}}\in D^{o}$, the proposed routing guidance algorithm can stabilize the total network queue length.*

Proof. Define the Lyapunov function as $\frac{1}{2}|X(t)|^2$, and the Lyapunov drift is $\frac{1}{2}|X(t+1)|^2-\frac{1}{2}|X(t)|^2$. Let $\Delta_{ij}(t)$ be the difference between $x_{ij}(t)$ and $x_{ij}(t)$.

$$\Delta_{ij}=x_{ij}(t+1)-x_{ij}(t)=-y_{ij}(t)+\sum_{h\in\Gamma_i^-}y_{hi}(t)p_{ij}(t)\quad\forall i\in\mathcal{L}_{\text{int}},j\in\Gamma_i^+ \tag{8.10}$$

$$\Delta_{ij}=-y_{ij}(t)+d_{ij}(t)\quad\forall i\in\mathcal{L}_{\text{entry}},j\in\Gamma_i^+ \tag{8.11}$$

$$\frac{1}{2}|X(t+1)|^2-\frac{1}{2}|X(t)|^2=\frac{1}{2}|X(t)+\Delta|^2-\frac{1}{2}|X(t)|^2=X(t)^{\intercal}\Delta+\frac{1}{2}|\Delta|^2 \tag{8.12}$$

$$|\Delta|^2=\sum_{i\in\mathcal{L}_{\text{int}}}|-y_{ij}(t)+\sum_{h\in\Gamma_i^-}y_{hi}(t)p_{ij}(t)|^2+\sum_{i\in\mathcal{L}_{\text{entry}}}|-y_{hi}(t)+d_i(t)p_{ij}(t)|^2 \tag{8.13}$$

The absolute value of Δ is bounded by the max values of capacity $\hat{Q}$ and demand rate $\hat{d}$. $\hat{Q}$ is the max capacity for all movements, $\hat{Q} = \max\{Q_{ij}, \forall (i,j) \in \mathcal{M}\}$. $\hat{Q}_{ij}$ is the maximum capacity for movement (i,j) in all phases, $\hat{Q}_{ij} = \left\{Q_{\phi}^{ij}, \forall \phi \in \Phi\right\}$

$$\begin{aligned} |-y_{ij}(t) + \sum_{h \in \Gamma_i^-} y_{hi}(t) p_{ij}(t)| &\leq \max\left(y_{ij}(t), \sum_{h \in \Gamma_i^-} y_{hi}(t) p_{ij}(t)\right) \\ &\leq \max\left(\hat{Q}_{ij}, \sum_{h \in \Gamma_i^-} \hat{Q}_{hi}\right) \leq \sum_{h \in \Gamma_i^-} \hat{Q} \end{aligned} \tag{8.14}$$

$$|-y_{ij}(t) + d_{ij}(t)| \leq \max\left(y_{ij}(t), d_{ij}(t)\right) \leq \max\left(\hat{Q}_{ij}, \hat{d}_{ij}\right) \leq \max\left(\hat{Q}, \hat{d}\right) \tag{8.15}$$

Let K be the upper bound of $\max\left(\sum_{h \in \Gamma_i^-} \hat{Q}, \hat{d}\right)$. Then $\Delta_{ij}(t) \leq K$. Let $|\mathcal{M}|$ denote the total number of movements in the network, then $|\Delta|^2 \leq |\mathcal{M}| K^2$.

$$\begin{aligned} X(t)^{\intercal}\Delta = &-\sum_{i \in \mathcal{L}} \sum_{j \in \Gamma_i^+} x_{ij}(t) y_{ij}(t) + \sum_{h \in \Gamma_i^-} \sum_{i \in \mathcal{L}} \sum_{j \in \Gamma_i^+} x_{ij}(t) y_{hi}(t) p_{ij}(t) \\ &- \sum_{i \in \mathcal{L}_{\text{entry}}} \sum_{j \in \Gamma_i^+} x_{ij}(t) y_{ij}(t) + \sum_{i \in \mathcal{L}_{\text{entry}}} x_{ij}(t) d_{ij}(t) \end{aligned} \tag{8.16}$$

$$\begin{aligned} X(t)^{\intercal}\Delta = &\sum_{i \in \mathcal{L}} \sum_{j \in \Gamma_i^+} x_{ij}(t) \left(-y_{ij}(t) + \sum_{h \in \Gamma_i^-} y_{hi}(t) p_{ij}(t)\right) \\ &+ \sum_{i \in \mathcal{L}_{\text{entry}}} \sum_{j \in \Gamma_i^+} x_{ij}(t) \left(-y_{ij}(t) + d_{ij}(t)\right) \end{aligned} \tag{8.17}$$

$$\begin{aligned} X(t)^{\intercal}\Delta = &\sum_{i \in \mathcal{L}} \sum_{j \in \Gamma_i^+} x_{ij}(t) \left(-y_{ij}(t) + \sum_{h \in \Gamma_i^-} y_{hi}(t) p_{ij}(t)\right) \\ &+ \sum_{i \in \mathcal{L}_{\text{entry}}} \sum_{j \in \Gamma_i^+} x_{ij}(t) \left(-y_{ij}(t) + d_{ij}(t)\right) \end{aligned}$$

$$= \sum_{i \in \mathcal{L}_{\text{entry}}} \sum_{j \in \Gamma_i^+} y_{ij}(t) \left(-x_{ij}(t) + \sum_{j \in \Gamma_i^+} x_{jk}(t) p_{jk}(t) \right) + \sum_{i \in \mathcal{L}_{\text{entry}}} \sum_{j \in \Gamma_i^+} x_{ij}(t) d_{ij}(t) \quad (8.18)$$

As $y_{ij} = \min\left(\sum_\phi Q_\phi^{ij} s_\phi(t), x_{ij}(t)\right)$, the upper bound of the term $y_{ij}\left(-x_{ij}(t) + \sum_{j \in \Gamma_i^+} x_{jk}(t) p_{jk}(t)\right)$ can be derived.

If $\sum_\phi Q_\phi^{ij} s_\phi(t) \leq x_{ij}(t)$, then:

$$-x_{ij}(t) \min\left(\sum_\phi Q_\phi^{ij} s_\phi(t), x_{ij}(t) \right) \leq -x_{ij}(t) \left(\sum_\phi Q_\phi^{ij} s_\phi(t) \right) \quad (8.19)$$

If $\sum_\phi Q_\phi^{ij} s_\phi(t) > x_{ij}(t)$, then:

$$\begin{aligned} &- x_{ij}(t) \min\left(\sum_\phi Q_\phi^{ij} s_\phi(t), x_{ij}(t) \right) < -x_{ij}^2 \\ &< -x_{ij}(t) \sum_\phi Q_\phi^{ij} s_\phi(t) + \left(\sum_\phi Q_\phi^{ij} s_\phi(t) \right)^2 \\ &< -x_{ij}(t) \left(\sum_\phi Q_\phi^{ij} s_\phi(t) \right) + \hat{Q}_{ij}^2 \end{aligned} \quad (8.20)$$

$\hat{Q}_{ij}$ is the maximum saturation flow for movement (i, j) in any phase. We also have:

$$y_{ij}(t) \sum_{j \in \Gamma_i^+} x_{jk}(t) p_{jk}(t) \leq \left(\sum_\phi Q_\phi^{ij} s_\phi(t) \right) \sum_{j \in \Gamma_i^+} x_{jk}(t) p_{jk}(t) \quad (8.21)$$

Then we can get:

$$
\begin{aligned}
X(t)^{\intercal}\Delta &\leq \sum_{i\in\mathcal{L}}\sum_{j\in\Gamma_i^+}\left(\sum_{\phi}Q_{\phi}^{ij}s_{\phi}(t)\right)\left(-x_{ij}(t)+\sum_{j\in\Gamma_i^+}x_{jk}(t)p_{jk}(t)\right)\\
&+\sum_{\mathcal{L}_{\text{entry}}}\sum_{j\in\Gamma_i^+}x_{ij}(t)d_{ij}(t)+\sum_{i\in\mathcal{L}}\sum_{j\in\Gamma_i^+}\hat{Q}_{ij}^2\\
&=\sum_{i\in\mathcal{L}}\sum_{j\in\Gamma_i^+}x_{ij}(t)\left(-\sum_{\phi}Q_{\phi}^{ij}s_{\phi}(t)+\sum_{h\in\Gamma_i^-}\sum_{\phi}Q_{\phi}^{hi}s_{\phi}(t)p_{ij}(t)\right)\\
&\qquad+\sum_{i\in\mathcal{L}_{\text{entry}}}\sum_{j\in\Gamma_i^+}x_{ij}(t)\left(d_{ij}(t)-\sum_{\phi}Q_{\phi}^{ij}s_{\phi}(t)\right)+\sum_{i\in\mathcal{L}}\sum_{j\in\Gamma_i^+}\hat{Q}_{ij}^2 \qquad (8.22)
\end{aligned}
$$

As it is assumed that $\bar{\mathbf{d}} \in D^o$, according to the stability region defined in equations (8.6), we have $\bar{y}_{ij} = \bar{d}_{ij} \leq \sum_{\phi} Q_{\phi}^{ij}\bar{s}_{\phi}$ for entry links and $\sum_{h\in\Gamma_i^-}\bar{y}_{hi}\bar{p}_{ij} = \bar{y}_{ij} \leq \sum_{\phi} Q_{\phi}^{ij}\bar{s}_{\phi}$ for internal links. Then we can find a stationary control policy $\bar{\mathcal{C}}$ that corresponds to a turning proportion vector $\bar{\mathbf{p}}$ and a control vector $\bar{\mathbf{s}}$, and there also exists a average flow vector $\bar{\mathbf{y}}$, a positive value ϵ, which satisfy:

$$
\bar{d}_{ij}+\epsilon \leq \sum_{\phi}Q_{\phi}^{ij}\bar{s}_{\phi}, \qquad \forall i\in\mathcal{L}_{\text{entry}} \qquad (8.23a)
$$

$$
\sum_{\phi}Q_{\phi}^{ij}\bar{s}_{\phi}+\epsilon \leq \sum_{h\in\Gamma_i^-}\sum_{\phi}Q_{\phi}^{hi}\bar{s}_{\phi}\bar{p}_{ij}, \qquad \forall i\in\mathcal{L}_{\text{int}} \qquad (8.23b)
$$

Then we have:

$$
\begin{aligned}
\mathbb{E}\left[X(t)^{\intercal}\Delta|X(t)\right] &= \mathbb{E}\left[\sum_{i\in\mathcal{L}}\sum_{j\in\Gamma_i^+}x_{ij}(t)\left(-\sum_{\phi}Q_{\phi}^{ij}\bar{s}_{\phi}+\sum_{h\in\Gamma_i^-}\sum_{\phi}Q_{\phi}^{hi}\bar{s}_{\phi}\bar{p}_{ij}\right)|X(t)\right]\\
&+\mathbb{E}\left[\sum_{i\in\mathcal{L}_{\text{entry}}}\sum_{j\in\Gamma_i^+}\left(\bar{d}_{ij}-\sum_{\phi}Q_{\phi}^{ij}\bar{s}_{\phi}\right)|X(t)\right]+\sum_{i\in\mathcal{L}}\sum_{j\in\Gamma_i^+}\hat{Q}_{ij}^2\\
&\leq -\epsilon\mathbb{E}\left[\sum_{i\in\mathcal{L}}\sum_{j\in\Gamma_i^+}x_{ij}(t)|X(t)\right]+\sum_{i\in\mathcal{L}}\sum_{j\in\Gamma_i^+}\hat{Q}_{ij}^2 \qquad (8.24)
\end{aligned}
$$

In this stationary policy, the turning proportion for movement (i, j) is always set to $\bar{p}_{ij}$, and the percent of time that is used for activating phase ϕ is set to be $\bar{s}_\phi$.

When the proposed algorithm $\mathcal{C}^\star = \{S^\star, P^\star\}$ is used, the following term is maximized:

$$\max \sum_{\phi\in\Phi} s_\phi(t)w_\phi(t) = \sum_{\phi\in\Phi} s^\star_\phi(t)\left(\sum_{(i,j)\in\phi} Q^{ij}_\phi\left(x_{ij}(t) - \sum_{j\in\Gamma^+_i} p^\star_{jk}(t)x_{jk}(t)\right)\right) = \sum_{(i,j)\in\mathcal{M}}\left(\sum_{\phi\in\Phi} Q^{ij}_\phi s^\star_\phi(t)\right)\left(x_{ij}(t) - \sum_{j\in\Gamma^+_i} p^\star_{jk}(t)x_{jk}(t)\right) \quad (8.25)$$

As

$$\sum_{(i,j)\in\mathcal{M}}\left(\sum_{\phi\in\Phi} Q^{ij}_\phi s^\star_\phi(t)\right)\left(x_{ij}(t) - \sum_{j\in\Gamma^+_i} p^\star_{jk}(t)x_{jk}(t)\right) \geq \sum_{(i,j)\in\mathcal{M}}\left(\sum_{\phi\in\Phi} Q^{ij}_\phi \bar{s}_\phi\right)\left(x_{ij}(t) - \sum_{j\in\Gamma^+_i} \bar{p}_{jk}x_{jk}(t)\right), \quad (8.26)$$

then we can get:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i\in\mathcal{L}}\sum_{j\in\Gamma^+_i}\left(\sum_{\phi} Q^{ij}_\phi s^\star_\phi(t)\right)\left(-x_{ij}(t) + \sum_{j\in\Gamma^+_i} x_{jk}(t)p^\star_{jk}(t)\right)|X(t), \mathcal{C}^\star\right] \\
&+ \mathbb{E}\left[\sum_{\mathcal{L}_{\text{entry}}}\sum_{j\in\Gamma^+_i} x_{ij}(t)\bar{d}_{ij}|X(t), \mathcal{C}^\star\right] \\
&\leq \mathbb{E}\left[\sum_{i\in\mathcal{L}}\sum_{j\in\Gamma^+_i}\left(\sum_{\phi} Q^{ij}_\phi \bar{s}_\phi\right)\left(-x_{ij}(t) + \sum_{j\in\Gamma^+_i} x_{jk}(t)\bar{p}_{jk}\right)|X(t), \bar{\mathcal{C}}\right] \\
&\qquad + \mathbb{E}\left[\sum_{\mathcal{L}_{\text{entry}}}\sum_{j\in\Gamma^+_i} x_{ij}(t)\bar{d}_{ij}|X(t), \bar{\mathcal{C}}\right] \quad (8.27)
\end{aligned}$$

Therefore, the upper bound used for the stationary control $\bar{\mathcal{C}}$ can also be used as the the upper bound for the proposed control $\mathcal{C}^\star$. The upper bound is:

$$\begin{aligned}
&\frac{1}{2}\mathbb{E}\left[|X(t+1)|^2 - |X(t)|^2|X(t)\right] = \mathbb{E}\left[X(t)^\intercal \Delta|X(t)\right] + \frac{1}{2}\mathbb{E}\left[|\Delta|^2|X(t)\right] \\
&\leq -\epsilon\mathbb{E}\left[\sum_{i\in\mathcal{L}_{\text{entry}}}\sum_{j\in\Gamma_i^+} x_{ij}(t)|X(t)\right] + \frac{1}{2}|\mathcal{M}|K^2 + \sum_{i\in\mathcal{L}}\sum_{j\in\Gamma_i^+}\hat{Q}_{ij}^2 \\
&\qquad\qquad = -\epsilon|X| + \frac{1}{2}|\mathcal{M}|K^2 + \sum_{i\in\mathcal{L}}\sum_{j\in\Gamma_i^+}\hat{Q}_{ij}^2 \quad (8.28)
\end{aligned}$$

Let $\mathcal{K} = \frac{1}{2}|\mathcal{M}|K^2 + \sum_{i\in\mathcal{L}}\sum_{j\in\Gamma_i^+}\hat{Q}_{ij}^2$. We can get the summation of the Lyapunov drift from $t=0$ to $t=T-1$:

$$\begin{aligned}
&\sum_{t=0}^{T-1}\left(\frac{1}{2}\mathbb{E}\left[|X(t+1)|\right]^2 - \frac{1}{2}\mathbb{E}\left[|X(t)|\right]^2\right) \\
&\qquad\qquad = \frac{1}{2}\mathbb{E}\left[|X(T)|\right]^2 - \frac{1}{2}\mathbb{E}\left[|X(0)|\right]^2 \leq T\mathcal{K} - \epsilon T|X| \quad (8.29)
\end{aligned}$$

$$\lim_{T\to\infty}\mathbb{E}\left[|X(t)|\right] \leq \frac{KT}{2\epsilon T} + \frac{1}{2\epsilon T}\mathbb{E}\left[|X(0)|\right]^2 - \frac{1}{2\epsilon T}\mathbb{E}\left[|X(T)|\right]^2 \leq \frac{\mathcal{K}}{\epsilon} + \frac{1}{2\epsilon T}|X(0)|^2 \quad (8.30)$$

□

8.2 Dynamic routing

The model in Section 8.1.1 is a simplified version of the max-pressure control with dynamic routing. In Section 8.1.1, the model modifies p_{jk} by setting the routes of vehicles on link j. When this model is applied in reality, it is hard for vehicles to move to the right downstream link when they are assigned to a different movement because these vehicles are already on lanes for their original movements, so switching lanes may produce more delays. The

following program in equation (8.31) is proposed to control the routes of vehicles on link i which allow vehicles to move the right lane smoothly.

$$\max \quad \sum_{\phi\in\Phi} w_\phi(t)s_\phi(t) \tag{8.31a}$$

$$\text{s.t.} \quad s_\phi(t) \in \{0,1\}, \forall\, \phi \in \Phi \tag{8.31b}$$

$$\sum_{\phi\in\Phi} s_\phi(t) = 1 \tag{8.31c}$$

$$w_{ij}(t) = x_{ij}(t) - \sum_{k\in\Gamma_j^+} p_{jk}(t)x_{jk}(t) \tag{8.31d}$$

$$w_\phi(t) = \sum_{(i,j)\in\phi} Q_\phi^{ij} w_{ij}(t) \tag{8.31e}$$

$$p_{jk}(t) \geq 0 \tag{8.31f}$$

$$p_{jk}(t) = \frac{\sum_i x_{ij}(t)p_{ijk}(t) + x_{jk}(t)}{\sum_i x_{ij}(t) + x_j(t)}, \quad (i,j) \in \phi \tag{8.31g}$$

$$p_{ijk}(t) = \frac{\sum_{o\in\mathcal{D}_{ij}} \mathcal{F}_k^o(t)x_{ij}^o(t)p_{ijk}^o(t)}{x_{ij}(t)} \tag{8.31h}$$

$$\sum_{k\in\Gamma_j^+} p_{jk}(t) = 1 \tag{8.31i}$$

$$\sum_{k\in\Gamma_j^+} p_{ijk}(t) = 1 \tag{8.31j}$$

$$\sum_{k\in\Gamma_j^+} p_{ijk}^o(t) = 1 \tag{8.31k}$$

In program (8.31), more decision variables are introduced to control the routes of vehicles in the queues on entry links of the current intersection. p_{ijk} represents the proportion of vehicles in the queue of movement (i,j) that join queue (j,k) after arriving link j. p_{ijk}^o further divides the p_{ijk} by vehicles' destination, which represents the proportion of vehicles in the queue $(i.j)$ heading for destination o that join queue (j,k) after arriving link j. For the max-pressure control, p_{jk} is the variable to be controlled. By controlling the values

of p_{ijk} and p^o_{ijk}, the value of p_{jk} can be controlled indirectly by modifying the routing behaviors of vehicle in queue (i, j). The relation between p_{jk} and p_{ijk} is shown in constraint (8.31g). For the right-hand side, the denominator is the total number of all incoming flow to link j added by the number of existing vehicle on link j. The numerator is the total number of incoming vehicles that turn to link j after arrival added by the existing queue length of movement (j, k) on link j. If there are exclusive lanes for each moving direction on link j, the value of p_{ijk} can change the distribution of queue length on different lanes. The relation between p_{ijk} and p^o_{ijk} is calculated by constraint (8.31h). For the right-hand side, the denominator is the total queue length of movement (i, j). The numerator is the summation of the product of the feasible parameter $\mathcal{F}^o_k$, the queue length variable x^o_{ij}, and the routing variable p^o_{ijk}. The numerator is actually x_{ijk} which is the number of vehicles on queue (i, j) that will join queue (j, k) after arriving link j. This number can include vehicles with destinations, that is why the numerator is in the form of summation. The feasible parameter $\mathcal{F}^o_k$ equals either 0 or 1. If it is 0, a vehicle heading for destination o becomes farther from its destination after traversing link k. If it is 1, link k brings a vehicle closer to its destination o. The reason for including this parameter is to exclude links that are not efficient to vehicles. To get the value of feasible parameter $\mathcal{F}^o_k$, the difference of potentials η of the source node and the destination node of link k should be calculated first, as shown in equation (8.32). The source node and destination node of link k is represented by θ^-_k and θ^+_k. The shortest path travel cost from nodes θ^-_k and θ^+_k to the destination node o is denoted as $\tau^o_{\theta^-_k}(t)$ and $\tau^o_{\theta^+_k}(t)$ respectively. Real-time measurement is preferred for the shortest path travel cost. $u^o_k(t)$ is the potential difference. If u^o_k is non-negative, the feasible parameter is 1, as shown in equation (8.33).

$$u^o_k(t) = \tau^o_{\theta^-_k}(t) - \tau^o_{\theta^+_k}(t) \tag{8.32}$$

$$\mathcal{F}_k^o = \begin{cases} 1, & \text{if } u_k^o(t) \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{8.33}$$

8.3 Experiment with macroscopic traffic models

8.3.1 System optimal traffic assignment

To compare the performance of the proposed algorithm with the result of system optimal dynamic traffic assignment (SODTA), SODTA model proposed by Beard and Ziliaskopoulos (2006) are modified in this section. This work is an extension of the study by Ziliaskopoulos (2000), which is the first study on SODTA that using the Cell Transmission Model (Daganzo, 1994) to represent flow propagation.

This SODTA model optimizes the signal control and the route choice at the same time, so the result of this model provides the smallest total system travel time that a network can get. If the total system travel for the proposed algorithm in the last section can be close to that of SODTA model, then the proposed algorithm should have a great performance.

The network has different types of cells. For example, C_R, C_S, C_M, C_D, C_O, and C_I represent the sets of source cells, sink cells, merging cells, diverging cells, ordinary cells, and intersection cells respectively. The set of intersection cells C_I includes all upstream cells next to each intersection, which are usually the most downstream cell on each link. C is the set of all cells. A pair of cells make up a cell connector. A connector set is denoted by E. Connectors are also categorized based on the cells they connect to. For example, if a connector is connected to a source cell, then it is a source connector. E_R, E_S, E_M, E_D, and E_O represent the sets of source connectors, sink connectors, merging connectors, diverging connectors, and ordinary connectors respectively. E_ϕ is the set of connectors activated in phase ϕ. Φ is a phase set. $\Gamma^-(i)$ and $\Gamma^+(i)$ represent the sets of upstream cells and downstream cells of cell i.

This mixed linear program has four decision variables: x_i^{odt}, y_{ij}^{odt}, q_i^t, w_i^t. x_i^{odt} represents the number of vehicles in cell i at time t that comes from source cell o and goes to sink cell d. y_{ij}^{odt} represents the flow goes from cell i to cell j whose origin is source cell o and destination is sink cell d. q_{ij}^t is the maximum flow of connector (i, j) at time t. This variable is 0 when the connector is not activated by the intersection control. S_ϕ^t represents the activation of phase ϕ at time t. Q_i^t is the capacity of cell i, which is can be calculated using the capacity of the road section and the time step size Δt. Q_ϕ^{ij} is the capacity of connector (i, j) in phase ϕ.

$$\text{min} \quad \sum_{\forall o \in C_R} \sum_{\forall d \in C_S} \sum_{\forall t \in T} \sum_{\forall i \in C \setminus C_S} x_i^{odt} \tag{8.34a}$$

$$\text{s.t.} \quad x_i^{odt} - x_i^{odt-1} - \sum_{k \in \Gamma^-} y_{ki}^{odt-1} + \sum_{j \in \Gamma^+(i)} y_{ij}^{odt-1} = 0 \tag{8.34b}$$

$$\forall o \in C_R, \forall d \in C_S, \forall i \in C \setminus \{C_R, C_S\}, \forall t \in T$$

$$x_o^{odt} - x_o^{odt-1} + y_{oj}^{odt-1} = d_o^{odt-1} \tag{8.34c}$$

$$\forall j \in \Gamma^+(o), \forall o \in C_R, \forall d \in C_S, \forall t \in T$$

$$x_d^{odt} - x_d^{odt-1} - y_{jd}^{odt-1} = 0 \tag{8.34d}$$

$$\forall j \in \Gamma^-(d), \forall o \in C_R, \forall d \in C_S, \forall t \in T$$

$$y_{ij}^{odt} - x_i^{odt} \leq 0, \tag{8.34e}$$

$$\forall o \in C_R, \forall d \in C_S, \forall (i, j) \in E_O \cup E_R \cup E_S \cup E_M, \forall t \in T$$

$$\sum_{j \in \Gamma^-(i)} y_{ij}^{odt} - x_i^{odt} \leq 0, \quad \forall o \in C_R, \forall d \in C_S, \forall i \in C_D, \forall t \in T \tag{8.34f}$$

$$\sum_{o \in C_R} \sum_{d \in C_S} \sum_{j \in \Gamma^+(i)} y_{ij}^{odt} \leq Q_i^t, \quad \forall i \in C_D, \forall t \in T \tag{8.34g}$$

$$\sum_{o \in C_R} \sum_{d \in C_S} \sum_{i \in \Gamma^-(j)} y_{ij}^{odt} \leq Q_j^t, \quad \forall j \in C_M, \forall t \in T \tag{8.34h}$$

$$\sum_{o \in C_R} \sum_{d \in C_S} y_{ij}^{odt} + \sum_{o \in C_R} \sum_{d \in C_S} \delta_j^t x_j^{odt} \leq \delta_j^t N_j^t, \tag{8.34i}$$

$$\forall (i,j) \in E_O \cup E_R \cup E_D, \forall t \in T$$

$$\sum_{o \in C_R} \sum_{d \in C_S} \sum_{i \in \Gamma^{-1}(j)} y_{ij}^{odt} + \sum_{o \in C_R} \sum_{d \in C_S} \delta_j^t x_j^{odt} \leq \delta_j^t N_j^t, \tag{8.34j}$$

$$\forall j \in C_M, \forall t \in T$$

$$y_{ij}^{od0} = 0, \quad \forall (i,j) \in E, \forall o \in C_R, \forall d \in C_S \tag{8.34k}$$

$$x_i^{odt} \geq 0, \quad \forall i \in C, \forall o \in C_R, \forall d \in C_S, \forall t \in T \tag{8.34l}$$

$$y_{ij}^{odt} \geq 0, \quad \forall (i,j) \in E, \forall o \in C_R, \forall d \in C_S, \forall t \in T \tag{8.34m}$$

$$\sum_{o \in C_R} \sum_{d \in C_S} \sum_{j \in \Gamma^{+}(i)} y_{ij}^{odt} \leq q_{ij}^t, \quad \forall (i,j) \in E_\phi, \forall \phi \in \Phi, \forall t \in T \tag{8.34n}$$

$$q_{ij}^t = S_\phi^t Q_\phi^{ij}, \quad \forall (i,j) \in E_\phi, \forall \phi \in \Phi, \forall t \in T \tag{8.34o}$$

$$\sum_{\phi \in \Phi} S_\phi^t = 1 \tag{8.34p}$$

The objective function in equation (8.34) minimizes the term that adds up the number of vehicles x_i^{odt} in all cells (except sink cells) for all origin-destination pairs in all time steps. If this term is multiplied by the time step size δ_t, it represents the total travel time of all vehicles. For each vehicle, its travel time equals the total number of times it shows up in each cell timing the time step size, so the term in the objective function timing the time step size equals to the total number of times all vehicles show up in each cell timing the time step size, which is the total travel time for all vehicles. Whether including the time step size in the objective function does not affect the solution, so the objective function only includes the occupancy term x_i^{ijo} to simplify the model.

Constraints (8.34b) to (8.34d) show the flow conservation for cells. The cell occupancy at time t equals the cell occupancy at time $t-1$ plus the incoming flow from upstream cells and minus the outgoing flow to downstream cells. For resource cells, the term for incoming flow is replaced by the demand rate d_o^{odt-1}, which is the number of vehicles entering the network through cell o at time $t-1$. For the destination node, the cell occupancy at

time t equals the cell occupancy at time $t-1$ plus the incoming flows from upstream cells. Destination cells have all arrived vehicles so their occupancy values always increase with time, and this is the reason why the objective function does not include the occupancy of destination cells. Constraints (8.34e)-(8.34j) set upper bounds for inter-cell flows. In constraints (8.34e) and (8.34f), the total flow with the same origin and destination leaving cell i should be less or equal to the corresponding cell occupancy. Constraints (8.34g)-(8.34h) show that the total outgoing flow should be less or equal to the capacity of both upstream and downstream cells. Constraints (8.34i) and (8.34j) show that the total flow entering a downstream cell should be less or equal to the available space in the downstream cell. Constraints (8.34k) sets the value of all flows equal to 0 when $t = 0$. Constraint (8.34n) set the upper bound of the flow of each connector at intersections as q. q is the produce of the capacity and the phase activation, as shown in constraint (8.34o). It is assumed that the intersection uses split phasing, which means each approach of the intersection has its own exclusive phase. When an approach is activated, all movement directions from this approach are allowed. The number of phases of an intersection equals the approach number of this intersection. Constraint (8.34p) shows that only one phase can be activated every time for each intersection.

A grid network was used to test the performance of the proposed max-pressure control. The network has 20 nodes and 56 links. There are two types of nodes: zones and intersections. The green nodes are zones served as the destination or the origin of the demand. The yellow nodes are signalized intersections with split phasing. There are two types of links: centroid connectors and normal links. The link connecting a zone and a yellow node is a centroid connector with infinite capacity, which is used to send vehicles into the network or take vehicles out of the network. Normal links have a free-flow speed of 60 mph. Vehicles entering the centroid connector directly join the point queue. Each of signalized intersections $1, 2, 3$, and 4 is connected to one centroid connector. These nodes have three approaches but with only two phases, as vehicles on centroid connectors are always al-

Table 8.1: Test network information

Test network information	
Number of lanes per link	1
Link free-flow speed (mph)	60
Link saturation flow (vph)	2200
Time step size (sec)	6
Total simulation time (min)	60
Total cell numbers	152
Cell jam density (veh/cell)	11
Cell saturation flow (veh/cell)	3.6
Cell w/v ratio	0.5

lowed to enter the network. For nodes 1 to 16, each of them has four phases and each phase activates all movements from an approach.

To convert this network to a cell-based network to get the result of the SODTA model, a centroid connector is represented by a source cell or destination cell, which depends on the direction of the connector. Normal links are divided into three cells. CPLEX with the JAVA API is used to create the SODTA model and the simulation time is set to be one hour.

Let $d = \sum_{o \in C_R} \sum_{d \in C_S} \sum_t d_o^{odt}$ be the total demand. The total system travel time is calculated as $\text{TSTT} = \Delta \sum_{\forall o \in C_R} \sum_{\forall d \in C_S} \sum_{\forall t \in T} \sum_{\forall i \in C \setminus C_S} x_i^{odt}$. Then the average travel time can be calculated as TSTT/d. The average travel times are calculated with demand ranging from 100 to 1200 vehicles.

8.3.2 Agent-based simulation using simulator based on CTM

To compare with the result of system optimal, a CTM-based macroscopic simulator is used here. For this simulator, the time step size is set to be 6 seconds. For max-pressure control, two types of path assignment results are used. One is the path assignment of the user equilibrium, which is calculated from the result of the dynamic traffic assignment from the same network. In the result of user equilibrium, there are 8 paths used for each origin-

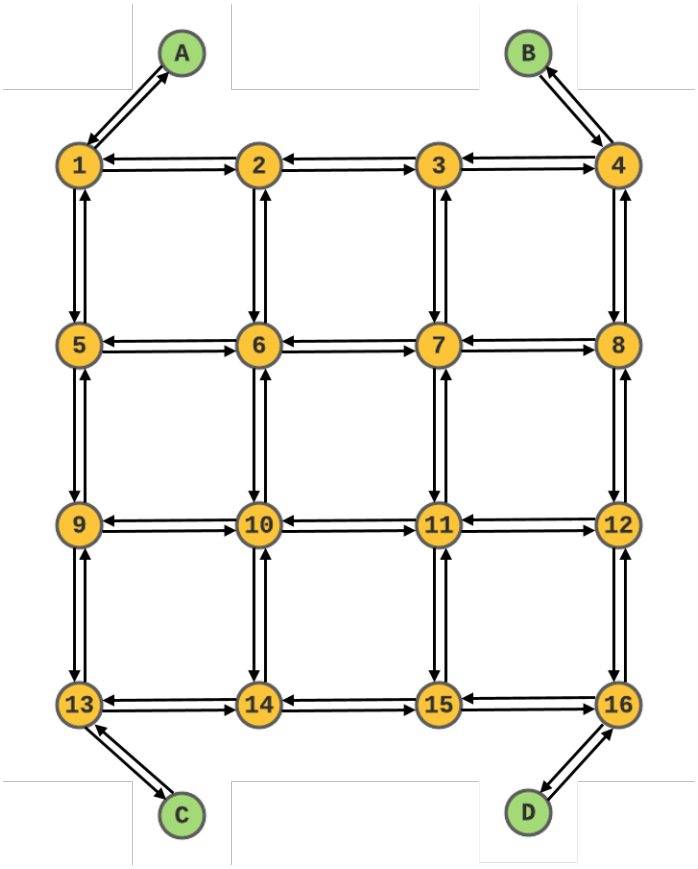


Figure 8.1: Test network for system optimal dynamic traffic assignment (SODTA)

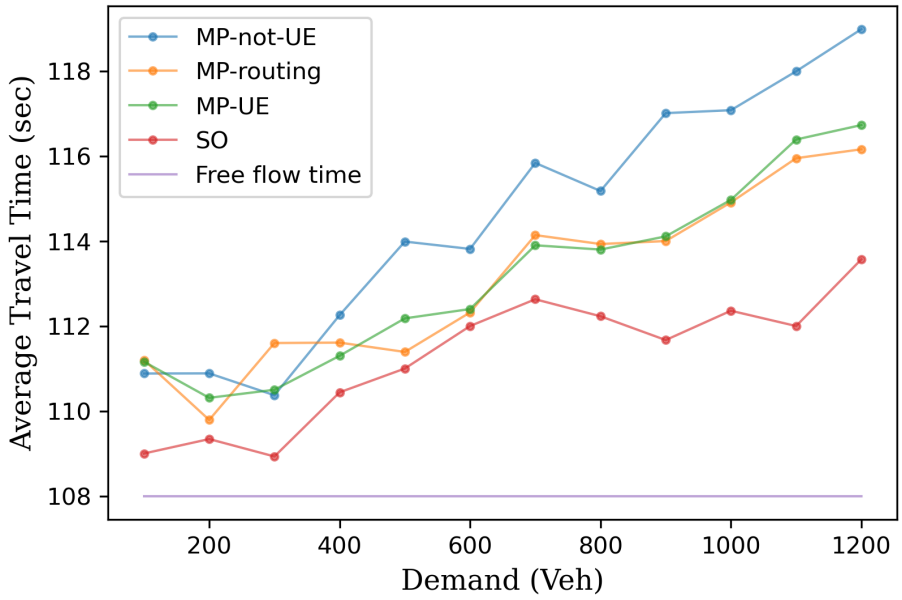


Figure 8.2: Comparison between average travel times

destination pair. Another one is the path assignment when the network is not under the user equilibrium. For this path assignment result, each origin-destination pair only uses two paths that have the largest probabilities in the path assignment result in user equilibrium. The tested demand ranges from 100 to 1200 vehicles.

Figure 8.2 shows the comparison of average travel times between system optimal condition (red line), the routing guidance algorithm (yellow line), and the max-pressure controls with UE path assignment and non-UE path assignment. The free flow travel time for the network is 108 seconds. Overall, there is a small difference between the results of system optimal and the max-pressure control results. The max-pressure control using the non-UE path assignment has the largest average travel time, which is 6 seconds larger than the result of system optimal on average. The routing guidance algorithm and the max-pressure control with the UE path assignment have similar average travel times, and when the demand is 1100 and 1200 vehicles, the routing guidance algorithm has a shorter average travel time than the max-pressure controller with the UE path assignment. Both controllers have aver-

age travel times that are larger than the system optimal result by less than 5 seconds. When the network is not congested, there is no large difference in travel times of paths in the network. The routing guidance algorithm tends to route vehicles to downstream links with fewer vehicles, with may have the similar effect of making vehicles to choose the shortest path. As the computation of the model in equation (8.34) is time-consuming and needs a large memory usage, the largest demand tested is set to be 1200 vehicles, which cannot make the network congested. For testing the performance of the proposed algorithm under fully congested conditions, a microscopic simulator is used in the following section.

8.4 Experiments using microscopic simulation

Macroscopic traffic models are based on deterministic relationships of the density, flow, and speed of the traffic. Microscopic traffic models can better capture the inter-vehicle relationships and the behavior of each vehicle using car-following and lane-changing models. Besides, observing whether a gridlock will be generated is also an important aspect of evaluation algorithm performance. Using microscopic simulators, the generation of gridlock can be better observed. To test the performance of the proposed routing guidance algorithm in a more realistic network, the microscopic simulator created in Chapter 7 is used here. The test network has 158 zones, 506 intersections in total, 167 signalized intersections, and 1186 links. The simulation time of each scenario is set to be 3 hours and vehicles enter the network in the first one and half hours .

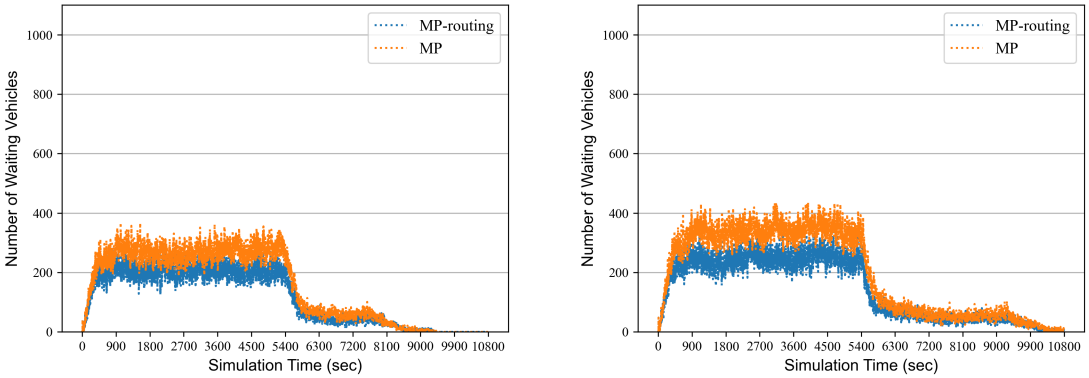
In this experiment, the probabilities of paths connecting each origin-destination pair are calculated by the result of dynamic traffic assignment. The turning proportions used by the max-pressure control are calculated based on the path probabilities and demand. A total demand of 48100 vehicles is used. The microscopic simulator SUMO has the option to set a percentage of the total demand to be used in the simulation. Here, we use 50%, 60%, 70%, and 80% of the total demand, which is equivalent to 16033, 19240, 22446, 25653

trips per hour respectively. The green time for the activated phase is set to be 15 seconds. The yellow time and the all-red time is set to be 3 seconds and 1 second. The mixed-integer programming proposed in equation (8.31) is solved with CPLEX at the end of the activated phase. The total number of waiting vehicles and the average travel time are collected during the simulation.

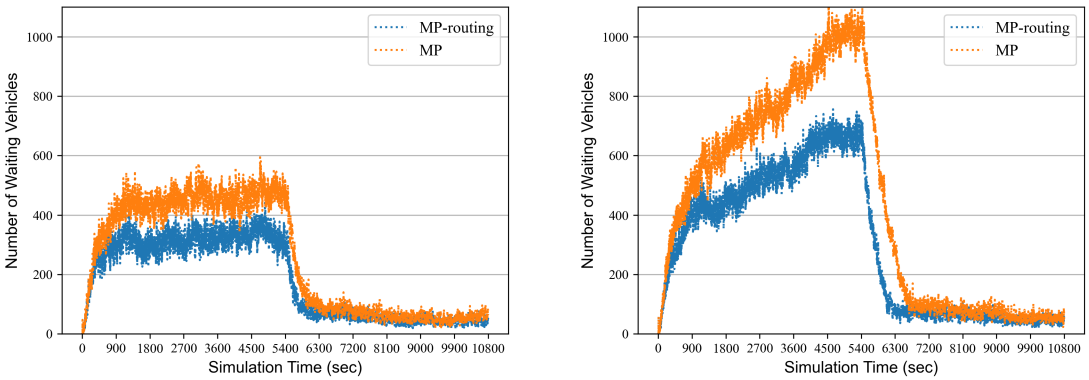
Figure 8.3 shows the comparison of the number of waiting for vehicles in the network with the proposed routing guidance algorithm and the max-pressure control proposed by Varaiya (2013) that uses movement queue length information to calculate intersection pressure. The simulator imports vehicle to the network from 0–5400 seconds. Vehicles that cannot enter the network will wait on entry links. The overall trend of the number of waiting vehicles is that it increases with a large rate starting from 0 seconds to 900 seconds. Then it stays at a constant if the controller can stabilize the queue length, otherwise, it keeps increasing until 5400 seconds. In the 50% and 60% demand scenarios, both controllers can stabilize the queue length while in the 70% and 80% demand scenarios, neither controller can stabilize the queue length. For four scenarios, the proposed routing guidance algorithm always has a smaller number of waiting vehicles than the normal max-pressure algorithm. Table 8.2 calculates the percentage of reduction in waiting vehicle number of the proposed algorithm compared to the normal max-pressure controller. It shows that the proposed algorithm reduces the number of waiting vehicles by at least 24.74% in these four scenarios. The percentage reduction also increases with the demand.

Figure 8.4 shows the comparison of average travel times for two controllers. The average travel time is constant when the queue length is stable between 15 minutes and 90 minutes and increases when the queue length is not stable. For all scenarios. then proposed algorithm has a shorter average travel time than the normal max-pressure algorithm. According to Table 8.2, the reduction in travel time of the proposed algorithm ranges from 6.79% to 16.13% and the reduction gets large when the network is more congested.

Figure 8.5 shows the comparison between the proposed algorithm and other types of



(a) Waiting vehicle numbers with 16033 trips per hour(b) Waiting vehicle numbers with 19240 trips per hour

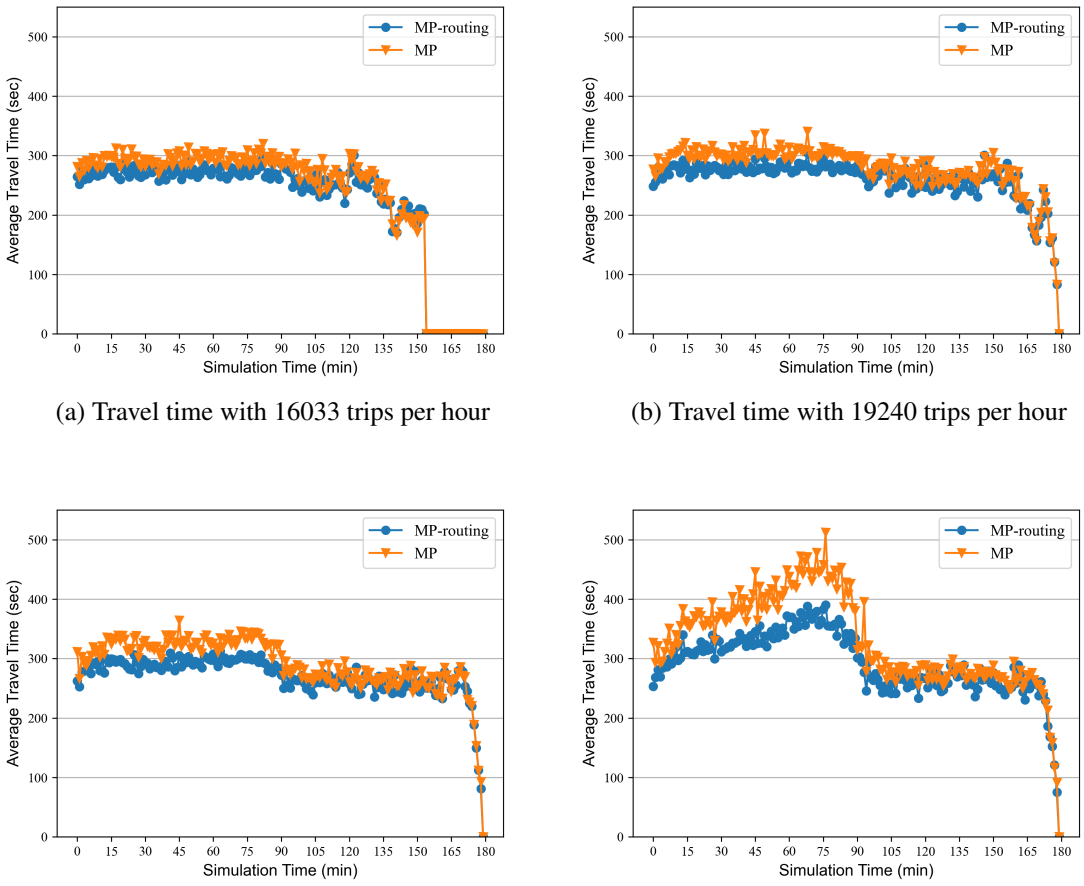


(c) Waiting vehicle numbers with 22446 trips per hour(d) Waiting vehicle numbers with 25653 trips per hour

Figure 8.3: Waiting vehicle numbers with different demands

Table 8.2: Comparison between MP-routing and MP controllers

Demand	Travel Time Reduction	Total Queue Length Reduction
16033 trips/hour	6.79 %	24.74 %
19240 trips/hour	8.12 %	27.96 %
22446 trips/hour	10.18 %	29.55 %
25653 trips/hour	16.13 %	30.72 %



(a) Travel time with 16033 trips per hour

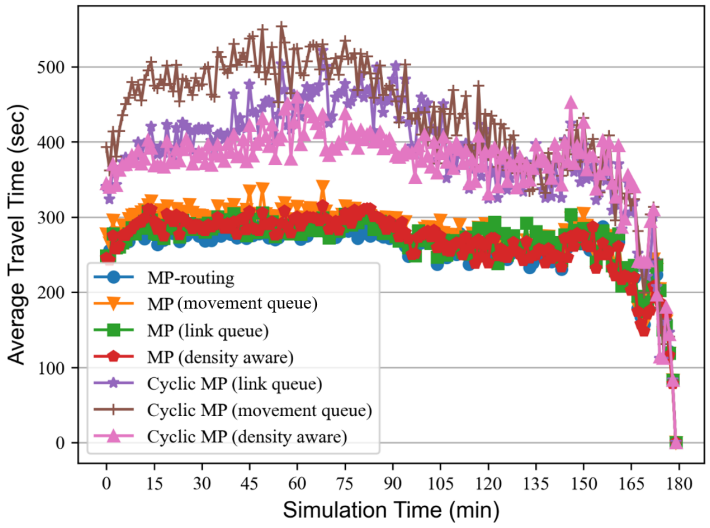
(b) Travel time with 19240 trips per hour

(c) Travel time with 22446 trips per hour

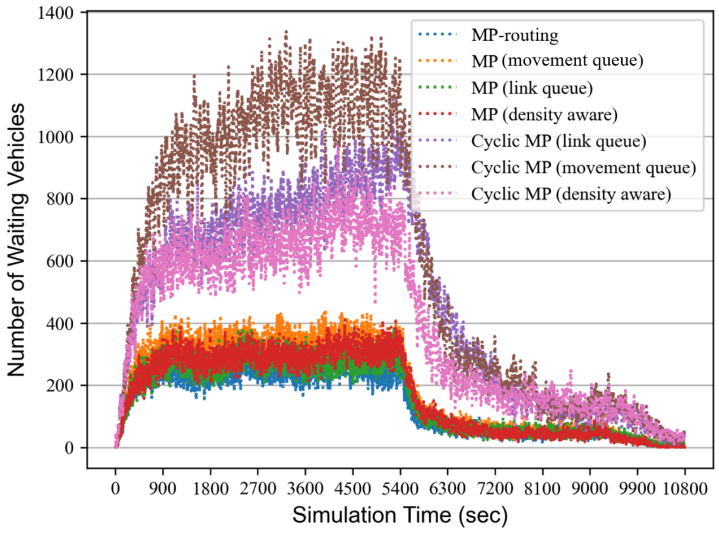
(d) Travel time with 25653 trips per hour

Figure 8.4: Travel times with different demands

max-pressure controllers. "MP-routing" refers to the proposed routing guidance algorithm in this chapter. "MP (movement queue)" and "MP (link queue)" refer to the max-pressure controllers of Varaiya (2013) and Wongpiromsarn et al. (2012) respectively. The former one uses a pressure function defined as $w_{ij} = x_{ij} - \sum_k x_{jk} p_{jk}$, and the latter one uses a pressure function defined as as $w_{ij} = x_i - x_j$. "MP (density aware)" is the density-aware back-pressure controller proposed by Gregoire et al. (2014b), which uses the information of the percent of occupied link space in their pressure function. All of the first four controllers are non-cyclic controllers, which means only the optimal phase is activated in every time step. The last three controllers are the cyclic version of "MP (movement queue)", "MP (link queue)", and "MP (density aware)" controllers respectively. After the calculation of phase pressure, the green time is assigned to each phase proportionally. These three cyclic controllers use a fixed cycle of 90 seconds. According to Figure 8.5, for the scenario with a demand of 19240 trips per hour, non-cyclic controllers have a smaller average travel time and a smaller number of waiting vehicles than cyclic controllers. The cyclic controllers will activate all phases even though some phases have a small weight, so part of the green time is wasted. The total number of waiting vehicles is stable under the control of the four non-cyclic controllers but cannot be stabilized by three cyclic controllers. Among four non-cyclic controllers, the proposed algorithm has the smallest number of waiting vehicles and average travel time.



(a) Travel time with 19240 trips per hour



(b) Waiting vehicle numbers with 60% demand

Figure 8.5: Comparison between types of max-pressure controllers

8.5 Conclusions

Intersection control and road users' route choices are two factors that affect network efficiency. Intersection control aims to reduce the travel time for the controlled area while road users seek their shortest paths to get to the destination. The route choice behaviors of road users may reduce the control effect of intersection controllers. The development of autonomous driving technology makes it possible to control road users' route choices, which helps to improve network performance. In this chapter, a routing guidance algorithm based on the max-pressure control is proposed. The stability property is proven for the routing guidance algorithm. The performance of the algorithm is first tested in a macroscopic traffic model and is compared to the results of system optimal route choice in the same network with a small demand. The proposed algorithm shows similar performance as the normal max-pressure controller with the user equilibrium path assignment. The performance of the proposed algorithm is then tested in a microscopic traffic simulator in which the path assignment of vehicles also follows the user equilibrium in a large network. The result shows that the proposed algorithm reduces the average time by at least 6.79% and reduces the number of waiting vehicles by at least 24.84% compared to the max-pressure controller of Varaiya (2013) in four test scenarios. The performances of these two controllers are also compared to the other five controllers, and the proposed routing guidance algorithm shows to have the best performance among all controllers.

Chapter 9

Conclusions

9.1 Research summary

Intersection control plays an important role in improving the efficiency of traffic networks and mitigate traffic congestion. A distributed traffic control system allows each intersection to operate by itself, which requires a less complex system structure and a shorter computation time than a centralized system when applied to a large network. This thesis uses a distributed control policy called the max-pressure control to control traffic. The max-pressure control was originally used as a scheduling policy in communication networks (Tassiulas and Ephremides, 1990) and was first applied in traffic networks in studies of Wongpiromsarn et al. (2012) and Varaiya (2013). Although the max-pressure control was proven to have some remarkable properties theoretically, some of its assumptions or requirements restrict its application in practice, such as fixed route choices and queue length information of movements. This thesis relaxes some assumptions of max-pressure, integrates it with signal-free algorithms to improve its applications, and analyzes its performances in various scenarios.

In Chapter 4, some mild assumptions of a travel time function are proposed to make the max-pressure control to maintain the network stability. The travel time function is assumed to be a monotone increasing function of the queue length with a bounded slope. In Chap-

ter 5, an autonomous intersection management algorithm considering pedestrians based on the max-pressure control is proposed. Activation of crosswalks is added to the model to allow pedestrian access. The model is proven to stabilize the network queue length. Experiments are used to explore the effects of traffic demand and pedestrian waiting time limit on intersection efficiency. The results show that an increase in vehicle demand results in an increase in pedestrian delays. When the vehicle demand is small, an increase in the demand decreases vehicle delays but increases vehicle delays when the demand is large. When the vehicle demand is much larger than the pedestrian demand, adding a small threshold for pedestrian waiting time to the model can significantly reduce pedestrian delays. In Chapter 6, dynamic traffic assignment is used to test the performance of max-pressure control when road users' route choices follow Wardrop's first principle. The convergence of the traffic assignment model is tested first. The stability region of the network changes in every iteration as the path assignment and turning proportions change in iterations. The model can converge when the demand is originally in the stability region, but it is not guaranteed to converge when the demand is far from the stability region. The average travel time and the total queue length of the network reduce with more iterations as the network is closer to the user equilibrium. Therefore, it is recommended to use the path assignment and the turning proportion corresponding to the result of user equilibrium when testing the effect of max-pressure control. In Chapter 7, a microscopic traffic simulator was built in SUMO with realistic settings. The performances of seven distributed controllers are tested in the simulator, including five max-pressure controllers using lane-based or link-based delay and queue length information, and two proportionally fair controllers using lane-based queue length and delay information. The result shows that the non-cyclic jam-aware max-pressure controller and the cyclic queue-based max-pressure controller have the best performances. In Chapter 8, a routing guidance algorithm based on the max-pressure control is proposed to modify road users' route choices. The proposed algorithm is proved to stabilize the network queue length if the demand is in the stability region. In the result of microscopic

simulation, the proposed algorithm has an average travel time that is at least 6% smaller and a total queue length that is at least 24% smaller than the max-pressure controller of Varaiya (2013).

9.2 Future work

Future work can be conducted in various directions.

1. Most existing studies prove the stability properties of the max-pressure control with point queue traffic flow models. Future work can incorporate more realistic traffic flow models in the proof, such as the cell transmission model. However, incorporating these models makes the proof much more complicated and significantly increases the difficulty of proving the stability.

2. In reality, the queue length information of each movement that is required for the max-pressure control is impossible to measure when there are combined lanes for multiple movements. The future work can be proposing max-pressure control algorithms using lane-based information because lane-based information is always measurable.

3. A simple extension of the current max-pressure control with AIM is incorporating multiple vehicle types when there are exclusive lanes, such as bicycles or buses.

References

Allsop, R. E. (1974). Some possibilities for using traffic control to influence trip distribution and route choice. In *Transportation and Traffic Theory, Proceedings*, volume 6.

Aziz, H. A. and Ukkusuri, S. V. (2012). Unified framework for dynamic traffic assignment and signal control with cell transmission model. *Transportation Research Record*, 2311(1):73–84.

Beard, C. and Ziliaskopoulos, A. (2006). System optimal signal optimization formulation. *Transportation Research Record*, 1978(1):102–112.

Bento, L. C., Parafita, R., and Nunes, U. (2012). Intelligent traffic management at intersections supported by V2V and V2I communications. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pages 1495–1502.

Bento, L. C., Parafita, R., Santos, S., and Nunes, U. (2013). Intelligent traffic management at intersections: Legacy mode for vehicles not equipped with v2v and v2i communications. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 726–731. IEEE.

Braess, D. (1968). Über ein paradoxon aus der verkehrsplanung. *Unternehmensforschung*, 12(1):258–268.

Bui, L., Srikant, R., and Stolyar, A. (2009). Novel architectures and algorithms for delay

reduction in back-pressure scheduling and routing. In *IEEE INFOCOM 2009*, pages 2936–2940. IEEE.

Chen, O. and Ben-Akiva, M. (1998). Game-theoretic formulations of interaction between dynamic traffic control and dynamic traffic assignment. *Transportation Research Record*, 1617(1):179–188.

Chen, R., Hu, J., Levin, M. W., and Rey, D. (2020). Stability-based analysis of autonomous intersection management with pedestrians. *Transportation Research Part C: Emerging Technologies*, 114:463–483.

Chiu, Y.-C., Bottom, J., Mahut, M., Paz, A., Balakrishna, R., Waller, T., and Hicks, J. (2011). Dynamic traffic assignment: A primer. *Transportation Research Circular*, (E-C153).

Daganzo, C. F. (1994). The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28(4):269–287.

De Campos, G. R., Falcone, P., Hult, R., Wymeersch, H., and Sjöberg, J. (2017). Traffic Coordination at Road Intersections: Autonomous Decision-Making Algorithms Using Model-Based Heuristics. *IEEE Intelligent Transportation Systems Magazine*, 9(1):8–21.

Dresner, K. and Stone, P. (2004). Multiagent traffic management: A reservation-based intersection control mechanism. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 530–537. IEEE Computer Society.

Dresner, K. and Stone, P. (2006). Traffic intersections of the future. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 1593. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Dresner, K. and Stone, P. (2007a). Learning policy selection for autonomous intersection management. In *The AAMAS 2007 Workshop on Adaptive and Learning Agents (ALAg 2007)*, pages 34–39.

Dresner, K. M. and Stone, P. (2007b). Sharing the road: Autonomous vehicles meet human drivers. In *IJCAI*, volume 7, pages 1263–1268.

Fajardo, D., Au, T.-C., Waller, S., Stone, P., and Yang, D. (2011). Automated intersection control: Performance of future innovation versus current traffic signal control. *Transportation Research Record: Journal of the Transportation Research Board*, (2259):223–232.

Fayazi, S. A., Vahidi, A., and Luckow, A. (2017). Optimal scheduling of autonomous vehicle arrivals at intelligent intersections via MILP. *Proceedings of the American Control Conference*, pages 4920–4925.

Garber, N. J. and Hoel, L. A. (2014). *Traffic and highway engineering*. Cengage Learning.

Gartner, N. H. (1983). *OPAC: A demand-responsive strategy for traffic signal control*. Number 906.

Gartner, N. H. and Al-Malik, M. (1996). Combined model for signal control and route choice in urban traffic networks. *Transportation Research Record*, 1554(1):27–35.

Gregoire, J., Frazzoli, E., de La Fortelle, A., and Wongpiromsarn, T. (2014a). Back-pressure traffic signal control with unknown routing rates. *IFAC Proceedings Volumes*, 47(3):11332–11337.

Gregoire, J., Qian, X., Frazzoli, E., De La Fortelle, A., and Wongpiromsarn, T. (2014b). Capacity-aware backpressure traffic signal control. *IEEE Transactions on Control of Network Systems*, 2(2):164–173.

Gregoire, J., Samaranayake, S., and Frazzoli, E. (2016). Back-pressure traffic signal control with partial routing control. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 6753–6758. IEEE.

Hamed, M. M. (2001). Analysis of pedestrians' behavior at pedestrian crossings. *Safety Science*, 38(1):63–82.

Hausknecht, M., Au, T.-C., and Stone, P. (2011). Autonomous intersection management: Multi-intersection optimization. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4581–4586. IEEE.

Hsieh, P.-C., Liu, X., Jiao, J., Hou, I.-H., Zhang, Y., and Kumar, P. (2017). Throughput-optimal scheduling for multi-hop networked transportation systems with switch-over delay. In *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 1–10.

Hunt, P., Robertson, D., Bretherton, R., and Winton, R. (1981). Scoot-a traffic responsive method of coordinating signals. Technical report.

Jin, Q., Wu, G., Boriboonsomsin, K., and Barth, M. (2012). Multi-agent intersection management for connected vehicles using an optimal scheduling approach. *Proceedings - 2012 International Conference on Connected Vehicles and Expo, ICCVE 2012*, pages 185–190.

Kamal, M. A., Imura, J., Ohata, A., Hayakawa, T., and Aihara, K. (2013). Coordination of automated vehicles at a traffic-lightless intersection. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, (Itsc):922–927.

Kamal, M. A. S., Imura, J. I., Hayakawa, T., Ohata, A., and Aihara, K. (2015). A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights. *IEEE Transactions on Intelligent Transportation Systems*, 16(3):1136–1147.

Kouvelas, A., Lioris, J., Fayazi, S. A., and Varaiya, P. (2014). Maximum pressure controller for stabilizing queues in signalized arterial networks. *Transportation Research Record: Journal of the Transportation Research Board*, 2421(1):133–141.

Le, T., Kovács, P., Walton, N., Vu, H. L., Andrew, L. L., and Hoogendoorn, S. S. (2015). Decentralized signal control for urban road networks. *Transportation Research Part C: Emerging Technologies*, 58:431–450.

Le, T., Vu, H. L., Walton, N., Hoogendoorn, S. P., Kovács, P., and Queija, R. N. (2017). Utility optimization framework for a distributed traffic control of urban road networks. *Transportation Research Part B: Methodological*, 105:539–558.

Levin, M. W. and Boyles, S. D. (2015). Intersection auctions and reservation-based control in dynamic traffic assignment. *Transportation Research Record: Journal of the Transportation Research Board*, (2497):35–44.

Levin, M. W. and Boyles, S. D. (2016). A multiclass cell transmission model for shared human and autonomous vehicle roads. *Transportation Research Part C: Emerging Technologies*, 62:103–116.

Levin, M. W. and Rey, D. (2017). Conflict-point formulation of intersection control for autonomous vehicles. *Transportation Research Part C: Emerging Technologies*, 85:528–547.

Li, B. (2013). A model of pedestrians' intended waiting times for street crossings at signalized intersections. *Transportation Research Part B: Methodological*, 51:17–28.

Li, L. and Jabari, S. E. (2019). Position weighted backpressure intersection control for urban networks. *Transportation Research Part B: Methodological*, 128:435–461.

Li, P., Mirchandani, P., and Zhou, X. (2015). Solving simultaneous route guidance and traffic signal optimization problem using space-phase-time hypernetwork. *Transportation Research Part B: Methodological*, 81:103–130.

Li, Z., Chitturi, M., Zheng, D., Bill, A., and Noyce, D. (2013). Modeling reservation-based autonomous intersection control in Vissim. *Transportation Research Record: Journal of the Transportation Research Board*, (2381):81–90.

Lioris, J., Kurzhanskiy, A., and Varaiya, P. (2016). Adaptive Max Pressure Control of Network of Signalized Intersections. *IFAC-PapersOnLine*, 49(22):19–24.

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and WieBner, E. (2018). Microscopic traffic simulation using sumo. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2575–2582. IEEE.

Ma, J., Li, X., Zhou, F., Hu, J., and Park, B. B. (2017). Parsimonious shooting heuristic for trajectory design of connected automated traffic part ii: computational issues and optimization. *Transportation Research Part B: Methodological*, 95:421–441.

Meneguzzer, C. (1995). An equilibrium route choice model with explicit treatment of the effect of intersections. *Transportation Research Part B: Methodological*, 29(5):329–356.

Mirchandani, P. and Head, L. (2001). A real-time traffic signal control system: architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies*, 9(6):415–432.

Mirheli, A., Hajibabai, L., and Hajbabaie, A. (2018). Development of a signal-head-free intersection control logic in a fully connected and autonomous vehicle environment. *Transportation Research Part C: Emerging Technologies*, 92:412–425.

Mirheli, A., Tajalli, M., Hajibabai, L., and Hajbabaie, A. (2019). A consensus-based distributed trajectory control in a signal-free intersection. *Transportation Research Part C: Emerging Technologies*, 100:161–176.

Neely, M. J. (2010). Stochastic network optimization with application to communication and queueing systems. *Synthesis Lectures on Communication Networks*, 3(1):1–211.

Pióro, M. and Medhi, D. (2004). Chapter 8 - fair networks. In Pióro, M. and Medhi, D., editors, *Routing, Flow, and Capacity Design in Communication and Computer Networks*, The Morgan Kaufmann Series in Networking, pages 307–349. Morgan Kaufmann, San Francisco.

Pumir, T., Anderson, L., Triantafyllos, D., and Bayen, A. M. (2015). Stability of modified max pressure controller with application to signalized traffic networks. In *2015 American Control Conference (ACC)*, pages 1879–1886. IEEE.

Pyzyk, K. (2018). Gridlock woes: Traffic congestion by the numbers. *Smart City Drive*.

Qian, X., Gregoire, J., Moutarde, F., and De La Fortelle, A. (2014). Priority-based coordination of autonomous and legacy vehicles at intersection. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1166–1171. IEEE.

Rey, D. and Levin, M. W. (2019). Blue phase: Optimal network traffic control for legacy and autonomous vehicles. *Transportation Research Part B: Methodological*, 130:105–129.

Sims, A. G. and Dobinson, K. W. (1980). The sydney coordinated adaptive traffic (scat) system philosophy and benefits. *IEEE Transactions on Vehicular Technology*, 29(2):130–137.

Smith, M. (1979). Traffic control and route-choice; a simple example. *Transportation Research Part B: Methodological*, 13(4):289–294.

Smith, M. (2015). Traffic signal control and route choice: A new assignment and control model which designs signal timings. *Transportation Research Part C: Emerging Technologies*, 58:451–473.

Smith, M. and Watling, D. (2016). A route-swapping dynamical system and lyapunov function for stochastic user equilibrium. *Transportation Research Part B: Methodological*, 85:132–141.

Sun, X. and Yin, Y. (2018). A simulation study on max pressure control of signalized intersections. *Transportation Research Record: Journal of the Transportation Research Board*, 2672(18):117–127.

Taale, H. (2008). Integrated anticipatory control of road networks: A game-theoretical approach.

Tassiulas, L. (1992). Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 31(12).

Tassiulas, L. and Ephremides, A. (1990). Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. In *29th IEEE Conference on Decision and Control*, pages 2130–2132. IEEE.

Ukkusuri, S., Doan, K., and Aziz, H. A. (2013). A bi-level formulation for the combined dynamic equilibrium based traffic signal control. *Procedia-Social and Behavioral Sciences*, 80:729–752.

Van den Berg, V. A. and Verhoef, E. T. (2016). Autonomous cars and dynamic bottleneck congestion: The effects on capacity, value of time and preference heterogeneity. *Transportation Research Part B: Methodological*, 94:43–60.

Van Kampen, J. (2015). Route guidance and signal control based on the back-pressure algorithm.

Varaiya, P. (2013). Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies*, 36:177–195.

Webster, F. V. (1958). Traffic signal settings. Technical report.

Wongpiromsarn, T., Uthaicharoenpong, T., Wang, Y., Frazzoli, E., and Wang, D. (2012). Distributed traffic signal control for maximum network throughput. In *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pages 588–595. IEEE.

Wu, J., Ghosal, D., Zhang, M., and Chuah, C.-N. (2017). Delay-based traffic signal control for throughput optimality and fairness at an isolated intersection. *IEEE Transactions on Vehicular Technology*, 67(2):896–909.

Wu, W., Zhang, J., Luo, A., and Cao, J. (2015). Distributed mutual exclusion algorithms for intersection traffic control. *IEEE Transactions on Parallel and Distributed Systems*, 26(1):65–74.

Wuthishuwong, C. and Traechtler, A. (2013). Vehicle to infrastructure based safe trajectory planning for autonomous intersection management. *2013 13th International Conference on ITS Telecommunications, ITST 2013*, pages 175–180.

Xia, C., Cochrane, C., DeGuire, J., Fan, G., Holmes, E., McGuirl, M., Murphy, P., Palmer, J., Carter, P., Slivinski, L., et al. (2017). Assimilating eulerian and lagrangian data in traffic-flow models. *Physica D: Nonlinear Phenomena*, 346:59–72.

Xiao, L. and Lo, H. K. (2015). Combined route choice and adaptive traffic control in a day-to-day dynamical system. *Networks and Spatial Economics*, 15(3):697–717.

Xiao, N., Frazzoli, E., Li, Y., Wang, Y., and Wang, D. (2014). Pressure releasing policy in traffic signal control with finite queue capacities. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 6492–6497.

Ying, L., Shakkottai, S., Reddy, A., and Liu, S. (2010). On combining shortest-path and back-pressure routing over multihop wireless networks. *IEEE/ACM Transactions on Networking*, 19(3):841–854.

Zaidi, A. A., Kulcsár, B., and Wymeersch, H. (2016). Back-pressure traffic signal control with fixed and adaptive routing for urban vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 17(8):2134–2143.

Zhang, K. and Nie, Y. M. (2018). Mitigating the impact of selfish routing: An optimal-ratio control scheme (orcs) inspired by autonomous driving. *Transportation Research Part C: Emerging Technologies*, 87:75–90.

Zhang, R., Li, Z., Feng, C., and Jiang, S. (2012). Traffic routing guidance algorithm based on backpressure with a trade-off between user satisfaction and traffic load. In *2012 IEEE Vehicular Technology Conference (VTC Fall)*, pages 1–5. IEEE.

Zhang, Y. J., Malikopoulos, A. A., and Cassandras, C. G. (2016). Optimal control and coordination of connected and automated vehicles at urban traffic intersections. In *2016 American Control Conference (ACC)*, pages 6227–6232. IEEE.

Zhu, F. and Ukkusuri, S. V. (2015). A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment. *Transportation Research Part C: Emerging Technologies*, 55:363–378.

Ziliaskopoulos, A. K. (2000). A linear programming model for the single destination system optimum dynamic traffic assignment problem. *Transportation Science*, 34(1):37–49.